



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

Larissa Santos Machado

Controle de um Modelo para Dengue

Recife

2017

Larissa Santos Machado

Controle de um Modelo para Dengue

Dissertação apresentada ao Programa de Pós-graduação em Matemática da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Mestrado em Matemática.

Orientador: César Augusto Rodrigues Castilho

Recife

2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M149c Machado, Larissa Santos
Controle de um modelo para dengue / Larissa Santos Machado. – 2017.
52 f.: il., fig., tab.

Orientador: César Augusto Rodrigues Castilho.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CCEN,
Matemática, Recife, 2017.
Inclui referências e apêndices.

1. Epidemias (matemática). 2. Biomatemática. I. Castilho, César Augusto
Rodrigues (orientador). II. Título.

519.85 CDD (23. ed.) UFPE- MEI 2017-60

LARISSA SANTOS MACHADO

CONTROLE DE UM MODELO PARA DENGUE

Dissertação apresentada ao Programa de Pós-graduação em Matemática da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Mestrado em Matemática.

Aprovada em: 21/02/2017.

BANCA EXAMINADORA

Prof. Dr. César Augusto Rodrigues Castilho(Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Airton Temistocles Gonçalves de Castro (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Renê Rodrigues Montenegro Filho (Examinador Externo)
Universidade Federal de Pernambuco

*A minha mãe Aldenôra
que quase infartou quando soube,
apenas cinco dias antes,
que eu iria morar em Recife.*

Agradecimentos

Soli Deo Gloria!

Agradeço a toda minha família - mãe (AAldy), pai (Marcos), irmã (Lay), vó (Marinês), tios (Gilvan e Vânia), madrinha (Verônica - A galega) e padrinho (Maurício) - que muito contribuiu e ajudou para que eu estivesse hoje concluindo mais esta etapa. Meus agradecimentos também a algumas amigas da família, Tetê e Vera, que foram de tamanha generosidade.

Aos professores que tanto me ensinaram, ajudaram de várias formas e me encorajaram, agradeço imensamente: meu orientador César Castilho e os profs. Hildeberto Cabral, Alan Almeida, Arlúcio Viana, Éder Mateus e Wagner Lemos.

Aos meus belos amigos sou muito grata pelas diversas festas propostas. Desculpa a ausência, eu estava estudando. :-D

Muito obrigada a todos por tudo!

Resumo

A Teoria de Controle Ótimo vem sendo aplicada a muitos problemas reais de diversas áreas. Sendo assim, neste trabalho, caracterizaremos Problemas de Controle Ótimo modelados a partir de Sistemas Dinâmicos formados por Equações Diferenciais Ordinárias e com isso provaremos o Princípio Máximo de Pontryagin que trabalha com a existência de soluções ótimas para os problemas introduzindo um conceito de Variável Adjunta. Com o intuito de obter soluções numéricas, conheceremos dois métodos de otimização (Método de Varredura Frente-Trás e Método do Gradiente) que serão aplicados de forma prática, com seus algoritmos escritos no MATLAB, em um exemplo tutorial de reação bioquímica. Após entendermos um pouco a teoria de controle, a nível epidemiológico, trabalharemos os conhecimentos obtidos em um modelo SIR (Suscetíveis - Infectados - Resistentes) aplicado à dengue e analisaremos o seu respectivo problema ótimo cujo objetivo é minimizar o número de indivíduos infectados utilizando técnicas de controle à mortalidade dos mosquitos, como campanhas de pulverização, em um cenário onde a temperatura ambiente tem em média 26°C . Para implementar este modelo, foram utilizados o software MATLAB com o Método de Varredura.

Palavras-chave: Problemas de Controle Ótimo. Princípio Máximo de Pontryagin. Métodos de Otimização. Dengue.

Abstract

The Optimal Control Theory has been applied to many real problems in various areas. Thus, in this work, we will characterize Optimal Control Problems modeled from Dynamic Systems formed by Ordinary Differential Equations and thus we will prove the Pontryagin's Maximum Principle that works with the existence of optimal solutions to the problems by introducing a concept of Adjoint Variable. In order to obtain numerical solutions, we will know two optimization methods (Forward-Backward Sweep Method and Gradient Method) that will be applied in a practical way, with their algorithms described in MATLAB, in a tutorial example of a biochemical reaction. After we get this theory of control, with regard to epidemiology, we will work the knowledge obtained in a SIR model (Susceptible - Infected - Resistant) applied to Dengue and we will analyze your respective optimal problem whose objective is to minimize the number of infected individuals using control techniques the mortality of mosquitoes, such as spraying campaigns, in a scenario where the mean temperature is 26°C. To implement this model, the MATLAB software and the Sweep Method were used.

Keywords: Optimal Control Problems. Pontryagin's Maximum Principle. Optimization Methods. Dengue.

Lista de ilustrações

Figura 1 – Modelo epidemiológico SIR. Fonte: (RODRIGUES et al., 2014)	35
Figura 2 – Fração do número de infectados Ih/Nh na presença de controle.	38
Figura 3 – Fração do número de infectados Ih/Nh na ausência de controle.	38
Figura 4 – Controle Ótimo $u^*(t)$ para a Figura 2.	39

Lista de tabelas

Tabela 1 – Parâmetros do sistema (3.0.1).	35
Tabela 2 – Valores dos parâmetros e condições iniciais do problema (P3). Fonte: (RODRIGUES; MONTEIRO; TORRES, 2014)	37

Sumário

1	INTRODUÇÃO	11
2	CONTROLE ÓTIMO	13
2.1	Problema Básico de Controle Ótimo	13
2.2	Princípio Máximo de Pontryagin	14
2.3	Métodos de Otimização	17
2.3.1	Método de Varredura Frente-Trás	17
2.3.2	Método do Gradiente	19
2.4	Um Problema de Reação Bioquímica	22
3	CONTROLE ÓTIMO APLICADO À DENGUE	33
4	CONSIDERAÇÕES FINAIS	40
	REFERÊNCIAS	41
	APÊNDICES	42
	APÊNDICE A –	43
A.1	O Espaço $L^2(\Omega)$	43
A.2	Definições	43
A.3	Interpretação da Adjunta	44
A.4	O Cone Normal	44
A.5	Runge-Kutta de Quarta Ordem (RK4)	45
	APÊNDICE B –	47
B.1	Controle Ótimo de Várias Variáveis	47
B.2	Programa desenvolvido para o problema P3	48

1 Introdução

A Teoria de Controle Ótimo corresponde a uma das técnicas de controle moderno utilizada em vários ramos do conhecimento. Cada vez mais difundida e estudada na física, biologia, economia, engenharia e na logística, esta técnica dispõe dos métodos da Otimização de funções para obter soluções "boas", que resolvam os chamados Problemas Ótimos. Estes problemas são modelados a partir dos respectivos sistemas dinâmicos que podem conter Equações Diferenciais Ordinárias, Diferenciais Parciais, Discretas, Estocásticas. Poucas são as vezes em que encontramos soluções explícitas para os problemas, sendo assim, uma maneira de obter a solução é aproximá-la de forma numérica. Então, temos que dispor de métodos numéricos que formulam algoritmos para o problema e de softwares que implementem os algoritmos criados.

De forma geral, um problema ótimo é escrito como

$$\max_u J(u, x) = \int_0^T f(t, u(t), x(t)) dt + \varphi(x(T)),$$

sujeito a

$$\begin{cases} x'(t) = g(t, u(t), x(t)), & t \in (0, T) \\ x(0) = x_0 \end{cases}$$

onde $J(u, x)$ é uma função, ou funcional, denominado de função objetivo, função custo, objetivo funcional ou custo funcional. Esse funcional geralmente é dado por uma integral cujo integrando são variáveis que caracterizam o sistema dinâmico. A variável estado, $x(t)$, descreve o comportamento do sistema e a variável controle, $u(t)$, entra como um controlador para o estado. Há diversas variedades de problemas ótimo: autônomos, não autônomos, com restrições na variável estado, com tempo final fixo ou livre, com controle limitado, com múltiplas variáveis e outros. No entanto, será abordado somente alguns desses mencionados. [(IZMAILOV A.; SOLODOV, 2014), (LENHART; WORKMAN, 2007)]

O objetivo deste texto é aplicar de maneira prática as ferramentas que a teoria de controle ótimo proporciona em um problema biológico referente à Dengue. Diante disso, mostrar como a introdução de controles nesse tipo de problema tem um resultado bastante positivo na redução do número de infectados. Neste texto, discutiremos apenas a teoria de controle ótimo para problemas autônomos com tempo final livre cujos sistemas correspondem a equações diferenciais ordinárias. O software utilizado para implementar os métodos será o MATLAB¹, desenvolvido pela MathWorks, formado por um sistema

¹ Para entender como este software é utilizado para resolver problemas ótimos, principalmente os deste texto, é recomendado consultar (ANITA; ARNAUTU; CAPASSO, 2011), Capítulo 1 e (LENHART; WORKMAN, 2007), Capítulo 5.

simples, mas muito eficiente, voltado para o cálculo numérico por meio de matrizes.

Os métodos de otimização abordados, como expõe o Capítulo 1, serão o Método de Varredura Frente-Trás, mais conhecido como Forward-Backward Sweep, e o Método do Gradiente ou de Descida Máxima. Também no mesmo capítulo veremos uma das ferramentas mais importantes dessa teoria que é o Princípio Máximo de Pontryagin e com isso um exemplo de como aplicá-los, tanto princípio quanto métodos. O Capítulo 2 apresentará um modelo matemático epidemiológico relativo à dengue. Embasado nesse modelo, um problema de controle ótimo será proposto e algumas simulações serão feitas utilizando o Método de Varredura visto no Capítulo 1.

No que se refere à epidemia citada, a Dengue é uma doença viral que atualmente preocupa muito o país. O número de casos registrados, e mesmo o de mortes, têm sido bastante elevado, principalmente em períodos de altas temperaturas que favorecem a proliferação do vetor da doença: o mosquito *aedes aegypti*. Seu ciclo de vida é curto e dividido em quatro estágios: ovo, larva, pupa e adulto. O vírus possui também quatro variações que resultam em diferentes tipos de dengue que são identificadas como: DEN-1, DEN-2, DEN-3 e DEN-4. Uma pessoa que contrai um desses sorotipos fica imune a ele, mas com os mesmos riscos de contrair um dos outros e caso aconteça um novo contágio há chances de que a doença evolua a um nível mais perigoso, como dengue hemorrágica. Além do temor da dengue, os vírus estão se tornando cada vez mais resistentes e se adaptando a ambientes que antes eram inatingíveis. Dessa forma, vemos o aparecimento de novas doenças como é o caso da febre Chikungunya e do Zika vírus, ambos transmitidos também pelo *aedes aegypti* e identificados no Brasil pela primeira vez em 2014 e 2015, respectivamente. Nesse pouco tempo, inúmeros e assustadores foram os casos registrados e as complicações causadas por essas doenças, principalmente em regiões onde os mosquitos se proliferam com facilidade. [(RODRIGUES; MONTEIRO; TORRES, 2014), (SAÚDE, b), (SAÚDE, a)] Diante desse cenário, várias formas de controle são estudadas; como controle da mortalidade do mosquito (uso de inseticidas), técnicas ecológicas (campanhas educacionais de conscientização da população); e com auxílio de ferramentas matemáticas, como a otimização, podemos obter as melhores formas de controle para diminuir a incidência dessas epidemias ou até mesmo erradicá-las.

O modelo epidemiológico presente no Capítulo 2 corresponde a um modelo SIR, isto é, os hospedeiros, neste caso os humanos, são classificados em três estados epidemiológicos: suscetíveis (S_h), infectados (I_h) e resistentes (R_h). Os resultados da aplicação do método numérico nesse modelo serão debatidos também no último capítulo. Nos Apêndices encontramos alguns conceitos necessários para o entendimento da teoria, o Apêndice A refere-se ao primeiro capítulo e o B ao segundo capítulo.

2 Controle Ótimo

O capítulo que se inicia apresentará em que consiste um problema de controle ótimo formado por equações diferenciais ordinárias, esclarecendo todos os seus termos presentes; trará de forma bastante objetiva a prova do Princípio Máximo de Pontryagin, o qual fornece um conjunto de condições necessárias de primeira ordem que determinam um controle ótimo para o problema. Além disso, abordará dois tipos de métodos para resolução de problemas ótimos por meio de uma aproximação da solução $u(t)$ desejada. Os métodos expostos serão o de Varredura Frente-Trás e o do Gradiente, ambos desenvolvidos em um caso particular do problema de controle ótimo. Em alguns passos de seus algoritmos, haverá a necessidade de utilizar algum método para resolução de EDOs e algum relacionado a buscas lineares. Para isso, trabalharemos com os algoritmos de Runge-Kutta e de Armijo, respectivamente. Por fim, na última seção, esses dois métodos serão aplicados em um problema de reação bioquímica como forma de exemplo.

2.1 Problema Básico de Controle Ótimo

Um problema básico de controle ótimo para um sistema de equações diferenciais ordinárias possui a seguinte representação, conforme (ANITA; ARNAUTU; CAPASSO, 2011):

$$\max_u J(u, x) = \int_0^T f(t, u(t), x(t)) dt + \varphi(x(T)), \quad (\text{P1})$$

sujeito a $u \in K \subset L^2(0, T; \mathbb{R}^m)$ e $T > 0$, onde x é solução do sistema

$$\begin{cases} x'(t) = g(t, u(t), x(t)), & t \in (0, T) \\ x(0) = x_0 \end{cases} \quad (2.1.1)$$

Neste contexto, por (ANITA; ARNAUTU; CAPASSO, 2011) e (LENHART; WORKMAN, 2007), cada termo é definido como sendo

- $L^2(0, T; \mathbb{R}^m)$, com $m \in \mathbb{N}^*$, o espaço das funções quadrado integráveis de $(0, T)$ em \mathbb{R}^m que representa o conjunto das funções controle;
- K subconjunto convexo fechado de $L^2(0, T; \mathbb{R}^m)$;
- $u \in K$ uma função contínua por partes;
- $x \in AC([0, T]; \mathbb{R}^n)$, com $n \in \mathbb{N}^*$, uma função diferenciável por partes onde $AC([0, T]; \mathbb{R}^n)$ é o espaço das funções absolutamente contínuas de $[0, T]$ em \mathbb{R}^n ;

- $f : [0, T] \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ e $g : [0, T] \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ funções continuamente diferenciáveis nos três argumentos;
- $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$;
- $u \mapsto J(u, x)$ o custo funcional.

O significado desses conceitos pode ser visto nos Apêndices A.1 e A.2. Assim, nosso problema básico de controle ótimo consiste em encontrar uma função $u(t)$ conhecida como função **controle** e sua função **estado** $x(t)$ correspondente para maximizar o **objetivo funcional** (P1) o qual vamos supor que possui uma única solução x para qualquer $u \in L^2(0, T; \mathbb{R}^m)$.

Definição 2.1. Dizemos que $u^*(t) \in K$ é um **controle ótimo** para o problema (P1) se

$$J(u^*, x^*) \geq J(u, x), \quad \forall u \in K.$$

O par (u^*, x^*) é chamado **par ótimo** e $J(u^*, x^*)$ é o **valor ótimo** do objetivo funcional.

Podemos observar que problemas de maximização podem ser transformados em problemas de minimização equivalentes fazendo apenas

$$\max_u \{J(u, x)\} = \min_u \{-J(u, x)\}$$

sujeito a $u \in K$.

2.2 Princípio Máximo de Pontryagin

Este princípio, desenvolvido pelo matemático russo Lev Pontryagin como um critério necessário para obter o melhor controle possível de um sistema dinâmico, traz a ideia de introduzir uma nova variável à equação diferencial da função objetivo. Conhecida como **variável adjunta** (Apêndice A.3), esta possui propriedades semelhantes aos Multiplicadores de Lagrange do cálculo de variações que maximizam ou minimizam funções na presença de restrições, que, neste caso, ocorrem sobre o controle ou o estado. De forma sucinta, o princípio, segundo (LENHART; WORKMAN, 2007), afirma que

Teorema 2.1. *Se $u^*(t)$ e $x^*(t)$ são ótimos para o problema (P1), então existe uma variável adjunta, diferenciável por partes, $\lambda(t)$ tal que*

$$H(t, u(t), x^*(t), \lambda(t)) \leq H(t, u^*(t), x^*(t), \lambda(t))$$

para todo controle u em cada tempo t , onde a função Hamiltoniana H é dada por

$$H(t, u(t), x(t), \lambda(t)) = f(t, u(t), x(t)) + \lambda(t)g(t, u(t), x(t))$$

e a adjunta $\lambda(t)$ satisfaz

$$\begin{aligned}\lambda'(t) &= -f_x(t, u^*(t), x^*(t)) - \lambda(t)g_x(t, u^*(t), x^*(t)), \\ \lambda(T) &= \varphi_x(x^*(T)).\end{aligned}$$

Prova: [(ANITA; ARNAUTU; CAPASSO, 2011) e (LENHART; WORKMAN, 2007)]

Suponha que $u^*(t)$ é um controle ótimo para (P1) e que $x^*(t)$ seja o seu estado correspondente, isto é,

$$\int_0^T f(t, u^*(t), x^*(t))dt + \varphi(x^*(T)) \geq \int_0^T f(t, u(t), x(t))dt + \varphi(x(T)),$$

para qualquer $u \in K$.

Assuma que as funções f , g , φ e a função definida em $L^2(0, T; \mathbb{R}^m)$ tal que $u \mapsto x$ são Gâteaux diferenciáveis (Apêndice A.2) em todo ponto. Aqui, todos os elementos de \mathbb{R}^n , para todo $n \in \mathbb{N}^*$, e todas as derivadas parciais de f e φ , isto é, f_u, f_x, φ_x , são considerados vetores colunas. Seja

$$V = \{v \in L^2(0, T; \mathbb{R}^m); u^* + \varepsilon v, \forall \varepsilon > 0 \text{ suficientemente pequeno}\}$$

o conjunto de funções-variação de u^* , contínuas por partes, tais que $u^* + \varepsilon v$ forma outro controle contínuo por partes. Seja $u^\varepsilon(t) = u^*(t) + \varepsilon v(t) \in K$, com $x^\varepsilon(t)$ o estado correspondente a $u^\varepsilon(t)$, isto é, $x^\varepsilon(t)$ satisfaz

$$\begin{cases} (x^\varepsilon)'(t) = g(t, u^\varepsilon(t), x^\varepsilon(t)), & t \in (0, T) \\ x^\varepsilon(0) = x_0. \end{cases}$$

Para qualquer $v \in V$, defina $z = \frac{\partial}{\partial \varepsilon} x^\varepsilon(v)$. Então, z é solução de

$$\begin{cases} z'(t) = g_u(t, u^*(t), x^*(t))v(t) + g_x(t, u^*(t), x^*(t))z(t), & t \in (0, T) \\ z(0) = 0. \end{cases} \quad (2.2.2)$$

Assim, para um $v \in V$ arbitrário porém fixo, temos

$$\int_0^T f(t, u^*(t), x^*(t))dt + \varphi(x^*(T)) \geq \int_0^T f(t, u^\varepsilon(t), x^\varepsilon(t))dt + \varphi(x^\varepsilon(T)).$$

Conseqüentemente,

$$\int_0^T \frac{1}{\varepsilon} [f(t, u^\varepsilon(t), x^\varepsilon(t)) - f(t, u^*(t), x^*(t))]dt + \frac{1}{\varepsilon} [\varphi(x^\varepsilon(T)) - \varphi(x^*(T))] \leq 0,$$

para quaisquer $v \in V$ e $\varepsilon > 0$ suficientemente pequeno. Passando o limite na inequação acima para $\varepsilon \rightarrow 0$ temos que

$$\begin{aligned} \int_0^T [v(t) \cdot f_u(t, u^*(t), x^*(t)) + z(t) \cdot f_x(t, u^*(t), x^*(t))]dt \\ + z(t) \cdot \varphi_x(x^*(T)) \leq 0, \end{aligned} \quad (2.2.3)$$

onde \cdot denota o produto escalar usual em \mathbb{R}^m e em \mathbb{R}^n . Agora, vamos introduzir uma função ou variável adjunta λ , diferenciável por partes no intervalo $[0, T]$. Suponha que λ é solução do problema adjunto (assumimos que essa solução existe e é única):

$$\begin{cases} \lambda'(t) = -f_x(t, u^*(t), x^*(t)) - g_x^*(t, u^*(t), x^*(t))\lambda(t), & t \in (0, T) \\ \lambda(T) = \varphi_x(x^*(T)), \end{cases} \quad (2.2.4)$$

onde g^* é o operador adjunto de g . Multiplicando (2.2.2) por λ e integrando por partes sobre $[0, T]$, temos

$$\begin{aligned} z(T) \cdot \lambda(T) - \int_0^T z(t) \cdot \lambda'(t) dt = \\ \int_0^T [g_u(t, u^*(t), x^*(t))v(t) + g_x(t, u^*(t), x^*(t))z(t)] \cdot \lambda(t) dt, \end{aligned}$$

para qualquer $v \in V$. Por (2.2.4) obtemos

$$\begin{aligned} z(T) \cdot \varphi_x(x^*(T)) + \int_0^T z(t) \cdot [f_x(t, u^*(t), x^*(t)) + g_x^*(t, u^*(t), x^*(t))\lambda(t)] dt = \\ \int_0^T [v(t) \cdot g_u^*(t, u^*(t), x^*(t))\lambda(t) + z(t) \cdot g_x^*(t, u^*(t), x^*(t))\lambda(t)] dt. \end{aligned}$$

Consequentemente

$$\begin{aligned} \int_0^T z(t) \cdot f_x(t, u^*(t), x^*(t)) dt + z(T) \cdot \varphi_x(x^*(T)) = \\ \int_0^T v(t) \cdot g_u^*(t, u^*(t), x^*(t))\lambda(t) dt, \end{aligned}$$

para qualquer $v \in V$. Por (2.2.3), chegamos que

$$\int_0^T v(t) \cdot [f_u(t, u^*(t), x^*(t)) + g_u^*(t, u^*(t), x^*(t))\lambda(t)] dt \leq 0,$$

para qualquer $v \in V$, que significa

$$f_u(\cdot, u^*, x^*) + \lambda g_u^*(\cdot, u^*, x^*) \in N_K(u^*), \quad (2.2.5)$$

onde $N_K(u^*)$ é o cone normal (Apêndice A.3) de K em u^* .

Definindo a função Hamiltoniana H por

$$H(t, u, x, \lambda) = f(t, u, x) + \lambda g(t, u, x),$$

obtemos, respectivamente, as mesmas condições de (2.1.1), (2.2.4) e (2.2.5) fazendo

$$x' = \frac{\partial H}{\partial \lambda}, \quad \lambda' = -\frac{\partial H}{\partial x}, \quad \text{e} \quad \frac{\partial H}{\partial u} \in N_K(u^*).$$

Observemos que esta última condição, similar a (2.2.5), pode ser satisfeita se fizermos $\partial H / \partial u = 0$. Então, neste texto, vamos considerar (2.2.5) como sendo

$$\frac{\partial H}{\partial u} = 0 \quad \text{para } u \in K,$$

para obtermos uma representação do controle $u(t)$. \square

As equações (2.1.1),(2.2.4),(2.2.5) representam o Princípio Máximo de Pontryagin e (2.2.4),(2.2.5) são as condições necessárias de otimalidade para o dado problema ótimo.

Para obter um controle ótimo u^* , precisamos, primeiramente, garantir a existência do mesmo, como supõe o Teorema 1.1. De acordo com (ANITA; ARNAUTU; CAPASSO, 2011), um método bem geral que garante a existência é:

Seja

$$d = \sup_{u \in K} J(u, x) \in \mathbb{R}.$$

Para qualquer $k \in \mathbb{N}^*$, existe $u_k \in K$, tal que

$$d - \frac{1}{k} < J(u_k, x_{u_k}) \leq d.$$

Passo 1: Prove que existe uma subsequência $\{u_{k_r}\}$ tal que

$$u_{k_r} \longrightarrow u^* \text{ fracamente em } L^2(0, T, \mathbb{R}^m).$$

Se K é limitado, a conclusão segue imediatamente. Como K é um subconjunto convexo fechado de $L^2(0, T, \mathbb{R}^m)$, K é também fracamente fechado e conseqüentemente $u^* \in K$.

Passo 2: Prove que existe uma subsequência $\{x_{u_{k_r}}\}$ que converge para x^* em $C([0, T], \mathbb{R}^n)$ (algumas vezes a convergência em $L^2(0, T, \mathbb{R}^n)$ é suficiente).

Passo 3: De

$$d - \frac{1}{k_r} < J(u_{k_r}, x_{u_{k_r}}) \leq d,$$

passando o limite obtemos $J(u^*, x^*) = d$. E, conseqüentemente, u^* é um controle ótimo para o problema (P1).

Em relação à existência e unicidade de soluções de problemas ótimos também podemos consultar (ARNAUTU; NEITTAANMAKI, 2003), Seção 2.1 e Apêndice A.3.

2.3 Métodos de Otimização

2.3.1 Método de Varredura Frente-Trás

Considere o seguinte caso particular do problema de controle ótimo (P1) onde $\varphi = 0$ para todo $t \in [0, T]$:

$$\max_u \int_0^T f(t, u(t), x(t)) dt,$$

sujeito a

$$x'(t) = g(t, u(t), x(t)), \quad x(0) = x_0;$$

cuja solução satisfaz

$$x'(t) = g(t, u(t), x(t)), \quad x(0) = x_0; \quad (2.3.6)$$

$$\lambda'(t) = -\frac{\partial H}{\partial x} = -f_x(t, u(t), x(t)) - \lambda(t)g_x(t, u(t), x(t)), \quad (2.3.7)$$

$$\lambda(T) = 0;$$

$$0 = \frac{\partial H}{\partial u} = f_u(t, u(t), x(t)) + \lambda(t)g_u(t, u(t), x(t)) \text{ em } u^*.$$

Com o intuito de encontrar uma solução numérica para o problema acima, vários algoritmos que geram uma aproximação para o controle ótimo $u^*(t)$ são propostos. Entre eles, o **Método de Varredura Frente-Trás** consiste em obter $u^*(t)$ através de uma sequência de iterações que percorrem o tempo para frente e para trás até convergirem à solução do problema.

Dado o intervalo $[0, T]$, inserimos uma quantia de N passos no intervalo $[0, T]$; $0 = b_1, b_2, \dots, b_N, b_{N+1} = T$; igualmente espaçados, e consideramos os vetores $\vec{u} = (u_1, u_2, \dots, u_{N+1})$, $\vec{x} = (x_1, x_2, \dots, x_{N+1})$, $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_{N+1})$ como sendo aproximações para o controle, o estado e a adjunta, respectivamente. Um esboço do algoritmo para esse método, conforme (LENHART; WORKMAN, 2007), segue abaixo:

S1: Atribua uma estimativa inicial para \vec{u} sobre o intervalo.

S2: Utilizando a condição inicial $x_1 = x(0) = x_0$ e os valores para \vec{u} , resolva \vec{x} , de acordo com sua equação diferencial presente em (2.3.6), avançando no tempo.

S3: Utilizando a condição de transversalidade $\lambda_{N+1} = \lambda(T) = 0$ e os valores para \vec{u} e \vec{x} , resolva $\vec{\lambda}$, de acordo com sua equação diferencial presente em (2.3.7), voltando no tempo.

S4: Atualize \vec{u} entrando com novos valores para \vec{x} e $\vec{\lambda}$ na caracterização do controle ótimo.

S5: Verifique a convergência. Se os valores das variáveis nesta iteração e na iteração anterior são suficientemente próximos, saia com os valores atuais como soluções. Se os valores não são próximos, retorne a S2.

Algumas observações sobre o algoritmo são necessárias. Como afirma (LENHART; WORKMAN, 2007), para a estimativa inicial em S1, $\vec{u} \equiv 0$ é quase sempre suficiente. Somente em alguns casos, onde ocorrem uma divisão por u , essa estimativa deve ser alterada. Para S2 e S3 qualquer método padrão de resolução de equações diferenciais pode ser utilizado. Trabalharemos, então, com o método de **Runge-Kutta de Quarta Ordem (RK4)** que consiste em obter uma solução aproximada do sistema de otimalidade comparando-a com um polinômio de Taylor apropriado a fim de eliminar o cálculo das derivadas de ordem superior (Apêndice A.5).

O erro para Runge-Kutta 4 e conseqüentemente para o método de varredura é $O(h^4)$. Para a convergência em S5, vamos usar um teste ligeiramente mais rigoroso, ou seja,

vamos exigir que o erro relativo seja suficientemente pequeno (LENHART; WORKMAN, 2007):

$$\frac{\|\vec{u} - \overrightarrow{oldu}\|}{\|\vec{u}\|} \leq \delta.$$

Onde \vec{u} é o vetor controle com os valores da iteração atual; \overrightarrow{oldu} é o vetor controle com valores da iteração anterior e $\|\cdot\|$ denota a norma l^1 para vetores. Multiplicando ambos os lados da inequação acima por $\|\vec{u}\|$, para retirá-lo do denominador, temos

$$\begin{aligned} \delta \|\vec{u}\| - \|\vec{u} - \overrightarrow{oldu}\| &\geq 0, \\ \delta \sum_{i=1}^{N+1} |u_i| - \sum_{i=1}^{N+1} |u_i - oldu_i| &\geq 0. \end{aligned}$$

Considerando $N = 1000$ e $\delta = 0.001$ no algoritmo que descreve a solução do PVI. Esse teste deve ser feito para todas as variáveis: u , x e λ .

2.3.2 Método do Gradiente

Consideremos um problema de controle ótimo similar a (P1), também com $\varphi = 0$, e todas as condições propostas para seus termos na Seção 1.1:

$$\max_u \Phi(u) = J(u, x) = \int_0^T f(t, u(t), x(t)) dt,$$

sujeito a $u \in K \subset L^2(0, T; \mathbb{R}^m) = U$ e $T > 0$, onde x é solução do sistema

$$\begin{cases} x'(t) = g(t, u(t), x(t)), & t \in (0, T) \\ x(0) = x_0 \end{cases} \quad (2.3.8)$$

Seguindo a prova do Princípio de Pontryagin feita no capítulo anterior, porém, removendo o termo $\varphi(x(T))$ em todos os passos, obtemos $\lambda(t)$ como solução do problema adjunto

$$\begin{cases} \lambda'(t) = -f_x(t, u^*(t), x^*(t)) - g_x^*(t, u^*(t), x^*(t))\lambda(t), & t \in (0, T) \\ \lambda(T) = 0, \end{cases} \quad (2.3.9)$$

e a função objetivo

$$\Phi_u(u) = f_u(\cdot, u, x) + \lambda g_u^*(\cdot, u, x). \quad (2.3.10)$$

Além disso, vimos também que qualquer problema de maximização pode ser visto e trabalhado como um problema de minimização, da forma

$$\min_u \Psi(u) = -J(u, x),$$

sujeito a $u \in K \subset U$, onde $\Psi(u) = -\Phi(u)$.

O **Método do Gradiente**, ou de **Máxima Descida**, é um tipo de algoritmo com buscas direcionais onde a cada iteração devemos encontrar uma direção ω a qual a função "diminui bastante" (ou "cresce bastante") e minimizá-la (ou maximizá-la) ao longo dessa direção. Assim, para este método, o sentido da direção tomada é $\omega = -\nabla\Phi(u)$ (ou $\omega = \nabla\Phi(u)$), ou seja, a direção oposta (ou semelhante) ao gradiente da função (BAUER, 2004).

Vamos, então, aplicar o método na função $\Phi(u)$. Em linhas gerais, este corresponde a: seja $u_k = u^{(k)}$ o controle atual, calcular sucessivamente a solução da equação estado $x^{(k)}$, a solução da equação adjunta $\lambda^{(k)}$, do gradiente $\Phi_u(u_k)$, que também podemos denotar por $\nabla\Phi(u_k)$, e o tamanho do passo $\rho \geq 0$ de forma a gerar uma maior aproximação da solução controle. O algoritmo correspondente para o problema de maximização, conforme (ANITA; ARNAUTU; CAPASSO, 2011), é:

E0: Escolha $u^{(0)} \in K$;

Faça $k := 0$.

E1: Calcule a solução $x^{(k)}$ para (2.3.8) sendo $u = u^{(k)}$:

$$\begin{cases} x'(t) = g(t, u^{(k)}(t), x(t)), & t \in (0, T) \\ x(0) = x_0 \end{cases}$$

E2: Calcule a solução $\lambda^{(k)}$ para (2.3.9) sendo $u := u^{(k)}$ e $x := x^{(k)}$:

$$\begin{cases} \lambda'(t) = -f_x(t, u^{(k)}(t), x^{(k)}(t)) - g_x^*(t, u^{(k)}(t), x^{(k)}(t))\lambda(t), & t \in (0, T) \\ \lambda(T) = 0, \end{cases}$$

E3: Calcule o gradiente $\omega^{(k)}$ usando a fórmula (2.3.10):

$$\omega^{(k)} := \Phi_u(u^{(k)}) = f_u(\cdot, u^{(k)}, x^{(k)}) + \lambda^{(k)} g_u^*(\cdot, u^{(k)}, x^{(k)}).$$

E4: Calcule o comprimento do passo $\rho_k \geq 0$ tal que

$$\Phi(P_K(u^{(k)} + \rho_k \omega^{(k)})) = \max_{\rho \geq 0} \Phi(P_K(u^{(k)} + \rho \omega^{(k)})).$$

(Este subproblema é chamado de *busca linear*.)

E5: $u^{(k+1)} := P_K(u^{(k)} + \rho_k \omega^{(k)})$.

E6: (O critério de parada)

Se $\|u^{(k+1)} - u^{(k)}\| < \varepsilon$

então PARE ($u^{(k+1)}$ é a aproximação do controle)

senão $k := k + 1$; vá para E1.

P_K é o operador projeção sobre o conjunto convexo K ; isto é, $P_K : U \rightarrow K$ definido por

$$\|P_K(u) - u\| \leq \|v - u\| \text{ para qualquer } v \in K,$$

onde $\|\cdot\|$ é a norma de U , $\rho > 0$ é o passo de descida da função e $\varepsilon > 0$ em E6 é uma precisão suficientemente pequena determinada.

Esclarecimentos e informes sobre o método do gradiente, a convergência do algoritmo acima e de outros algoritmos estão presentes em (ARNAUTU; NEITTAANMAKI, 2003), Capítulo 2.

Os passos E1 e E2 são obtidos a partir do método de Runge-Kutta de quarta ordem descrito na seção anterior 2.1, onde a equação estado é calculada percorrendo o tempo de 0 a T , e a equação adjunta é obtida decrescendo o tempo de T a 0. Em E4, a função projeção P_K é determinada, de acordo com cada problema, de modo a ser implementada em qualquer programa computacional. Além disso, o valor ρ_k é encontrado por uma aproximação da fórmula em E4, feita pelo **Método de Armijo**. Este corresponde a um dos métodos mais eficientes de busca linear, realiza uma pesquisa de modo sequencial, elemento por elemento, e tem como algoritmo (ANITA; ARNAUTU; CAPASSO, 2011):

A0: Seja $\bar{\rho} := \rho_{k-1}$.

A1: $u := P_K(u^{(k)} + \bar{\rho}\omega^{(k)})$.

A2: Calcule a solução x_1 do problema estado com entrada u :

$$\begin{cases} x_1'(t) = g(t, u(t), x(t)), & t \in (0, T) \\ x_1(0) = x_0 \end{cases}$$

A3: Calcule $J(u, x_1)$.

A4: (O critério de parada)

Se $J(u, x_1) \leq J(u^{(k)}, x^{(k)})$

então $\bar{\rho} := b\bar{\rho}$; vá para A1

senão $\rho_k := \bar{\rho}$; PARAR algoritmo.

No caso em que o subconjunto K é todo o espaço U , teremos $P_K = P_U = I_U$ e E4 se tornará

$$\Phi(u^{(k)} + \rho_k \omega^{(k)}) = \max_{\rho \geq 0} \Phi(u^{(k)} + \rho \omega^{(k)}).$$

Se o problema for de minimização, o algoritmo do método do gradiente terá apenas as seguintes mudanças:

- E3: tomamos $\omega^{(k)} := -\Phi_u(u^{(k)})$ para obter a direção de descida.
- E4: tomamos $\min_{\rho \geq 0}$ ao invés de $\max_{\rho \geq 0}$.

Um pouco da teoria sobre o método de descida máxima para problemas de minimização e a relação com problemas de maximização pode ser vista em (ANITA; ARNAUTU; CAPASSO, 2011), Seção 3.1. Sobre o método de Armijo e sua convergência podemos verificar (ARMIJO, 1966) e (ARNAUTU; NEITTAANMAKI, 2003).

2.4 Um Problema de Reação Bioquímica

A fim de compreender o desenvolvimento dos algoritmos de ambos os métodos, vamos aplicá-los, a nível de exemplo, em um mesmo problema de reação bioquímica, presente em (ANITA; ARNAUTU; CAPASSO, 2011), Seção 3.2, com o MATLAB como programa computacional utilizado. Iniciaremos com o algoritmo do gradiente e, logo após, o de varredura frente-trás.

Em uma reação bioquímica, um componente é adicionado a uma taxa constante em um intervalo de tempo $[0, T]$, com $T > 0$. Sejam $x(t)$ a variação do pH em um tempo $t \in [0, T]$ e $u(t)$ o controle desse pH, já que a qualidade do produto depende desse controle. O valor de $x(t)$ satisfaz o seguinte problema de valor inicial:

$$\begin{cases} x'(t) = \alpha x(t) + \beta u(t), & t \in (0, T) \\ x(0) = x_0, \end{cases} \quad (\text{P2})$$

onde α e β são constantes positivas e x_0 é a variação inicial do valor do pH. Supomos que a diminuição no rendimento do valor do pH devido a alterações nele é dada por $\int_0^T x(t)^2 dt$. Supomos também que a taxa de custo para manter $u(t)$ é proporcional a $u(t)^2$. Logo, o modelo bioquímico leva ao seguinte problema de controle ótimo

$$\min_u \frac{1}{2} \int_0^T [ax(t)^2 + u(t)^2] dt,$$

sujeito a $u \in U = L^2(0, T)$, onde x é solução para o PVI acima e a é um constante positiva. Esse problema é similar ao da seção (2.3.2) com $m = 1, n = 1$ e

$$\begin{aligned} f(t, u, x) &= -\frac{1}{2}(ax^2 + u^2) \\ g(t, u, x) &= \alpha x + \beta u. \end{aligned}$$

Considere

$$\begin{aligned} \Phi(u) &= -\frac{1}{2} \int_0^T [ax(t)^2 + u(t)^2] dt \quad e \\ \Psi(u) &= \frac{1}{2} \int_0^T [ax(t)^2 + u(t)^2] dt, \end{aligned}$$

assim, $\Psi(u) = -\Phi(u)$ e o problema ótimo pode ser reformulado como

$$\min_u \Psi(u) = \min_u (-\Phi(u)) = \max_u (\Phi(u)), \quad \text{para } u \in U.$$

Dessa forma, aplicando toda a teoria vista na prova do Princípio de Pontryagin, obtemos o gradiente do custo funcional:

$$\Psi_u(u^*) = u^* - \beta \lambda.$$

Escrevemos, então, o algoritmo do gradiente utilizando $-\Psi_u(u)$ como a direção de máxima descida.

Método do Gradiente para o problema (P2)**E0:** Escolha $u^{(0)} \in U$;Faça $k := 0$.**E1:** Calcule a solução $x^{(k)}$ como entrada $u := u^{(k)}$ para

$$\begin{cases} x'(t) = \alpha x(t) + \beta u^{(k)}(t), & t \in (0, T) \\ x(0) = x_0. \end{cases}$$

E2: Calcule a solução $\lambda^{(k)}$ com entrada $x := x^{(k)}$ para

$$\begin{cases} \lambda'(t) = -\alpha \lambda(t) + ax^{(k)}(t), & t \in (0, T) \\ \lambda(T) = 0. \end{cases}$$

E3: Calcule o gradiente $\omega^{(k)}$ usando

$$\omega^{(k)} := \Phi_u(u^{(k)}) = u^{(k)} - \beta \lambda.$$

E4: Calcule o comprimento do passo $\rho_k \geq 0$ tal que

$$\Phi(u^{(k)} - \rho_k \omega^{(k)}) = \min_{\rho \geq 0} \Phi(u^{(k)} - \rho \omega^{(k)}).$$

E5: Calcule o novo controle

$$u^{(k+1)} := u^{(k)} - \rho_k \omega^{(k)}.$$

E6: (O critério de parada)Se $\|u^{(k+1)} - u^{(k)}\| < \varepsilon$ então PARE ($u^{(k+1)}$ é a aproximação do controle)senão $k := k + 1$; vá para E1.

Vamos agora construir o programa para o algoritmo acima, conforme (ANITA; ARNAUTU; CAPASSO, 2011). Ele será dividido em sucessivas partes para uma melhor compreensão da implementação de seus passos. O programa inteiro pode ser obtido unindo todas as partes.

Considere o intervalo de tempo dividido equidistantemente da forma

$$t_i = (i - 1)h, \quad i = 1, \dots, n; \quad \text{onde } h = \frac{T}{n - 1},$$

e os vetores

$$uold = u^{(k)} \quad unew = u^{(k+1)}.$$

A PARTE 1 do programa contém as informações iniciais para o algoritmo, os dados de entrada e a implementação do passo E0, onde tudo após o sinal % corresponde apenas a uma descrição do comando.

PARTE 1

```

=====
clear
global alf bet ind
global a
global h
global uold
global u
global x
T = input('final time : '); % entrar com o tempo final
alf = input('alpha : '); % entrar com o valor de  $\alpha$ 
bet = input('beta : '); % entrar com o valor de  $\beta$ 
a = input('a : ');
x0 = input('x(0) : ');
h = input('h : '); % entrar com o tamanho do passo dado no tempo
t = 0:h:T; % intervalo de tempo com seu passo
t = t'; % muda para um vetor coluna
n = length(t);
x = zeros(n,1); % define o vetor estado e seu tamanho
x1 = zeros(n,1); % vetor estado a ser utilizado no loop de Armijo
p = lambda; % renomeando a variável adjunta  $\lambda$ 
p = zeros(n,1); % define o vetor adjunta
uold = zeros(n,1); % vetor controle correspondente a  $u^{(k)}$ 
unew = zeros(n,1); % vetor controle correspondente a  $u^{(k+1)}$ 
u = zeros(n,1); % vetor controle a ser utilizado no loop de Armijo
% Inicinado o passo E0:
u0 = input('u0 : ');
for i = 1:n
    uold(i) = u0;
end
disp('enter control data'); % introduzir dados do controle
eps = input('precision : '); % precisão  $\varepsilon$  para o algoritmo do gradiente
maxit = input('maxiter : '); % n° máx. de iterações para o algoritmo
disp('RO data');
roin = input('RO : '); % valor inicial para o passo  $\rho$  do gradiente
bro = input('b for RO : '); % valor do parâmetro  $b$  para o loop de  $\rho$ 
eps1 = input('precision for RO : '); % precisão  $\varepsilon_1$  para o passo  $\rho$ 
maxro = input('max for RO : '); % n° máx. de iterações para o loop de  $\rho$ 
ro = roin; % inicializando o  $\rho$ 
flag1 = 0; % indicador de convergência para o algoritmo do gradiente
ii = 0; % índice para o valor do gradiente

```

Os valores para os indicadores de convergência são:

0 - Nenhuma convergência foi obtida.

1 - Convergência foi obtida.

A variável *ii* guarda o índice do vetor *grad* responsável por armazenar o valor do gradiente para as sucessivas iterações. A PARTE 2 seguinte mostra o início do loop para o gradiente e o desenvolvimento do passo E1.

```

===== PARTE 2 =====
for iter = 1:maxit
    iter
    % E1: resolver a equação estado pelo método de Runge-Kutta
    x(1) = x0; % a condição inicial
    for i = 1:n
        ind = i; % variável de índice global utilizado pelo algoritmo de RK
        x(i+1) = RK41(x(i),t(i));
    end
    disp('SE solved'); % Equação Estado resolvida
    if iter == 1
        Q = a*x.^2 + uold.^2;
        temp = trapz(t,Q); % função que calcula o valor do funcional Q
        cvold = temp/2
        cost(1) = cvold; % armazena o valor do custo funcional
        jj = 1; % índice do vetor cost acima
    end
end

```

A equação estado no passo E1 é obtida pelo método de Runge-Kutta de ordem 4 que foi definido como RK41. Em um novo arquivo no editor do MATLAB, implementamos a função *RK41.m* seguinte forma:

```

===== RK41.m =====
function yout = RK41(x,t)
global h
tm = t + h/2;
k1 = h * F1(t,x);
k2 = h * F1(tm,x + k1/2);
k3 = h * F1(tm,x + k2/2);
k4 = h * F1(t + h,x + k3);
yout = x + (k1 + k4 + 2*(k2 + k3))/6;

```

onde a função *F1.m* correspondente é dada por

```
===== F1.m =====
function yout = F1(t,x)
global alf bet ind
global uold
yout = alf*x + bet*uold(ind);
```

Agora, fazemos os passos E2 e E3 do algoritmo que são resolver a equação adjunta também por Runge-Kutta, porém decrescendo o tempo e, assim, calcular o gradiente.

```
===== PARTE 3 =====
% E2: resolver a equação adjunta
p(n) = 0;
for j = 1:n-1
    i=n-j;
    ip1 = i+1;
    ind = ip1;
    p(i) = RK43(p(ip1),t(ip1));
end
disp('AE solved'); % Equação Adjunta resolvida
% E3: calcular o gradiente
w = uold-bet*p;
disp('GRADIENT COMPUTED'); % Gradiente Calculado
```

As funções *RK43.m* e *F3.m* para Runge-Kutta são respectivamente

```
===== RK43.m =====
function yout = RK43(q,t)
global h
h1 = -h;
tm = t + h1/2;
k1 = h1 * F3(t,q);
k2 = h1 * F3(tm,q + k1/2);
k3 = h1 * F3(tm,q + k2/2);
k4 = h1 * F3(t + h1,q + k3);
yout = q + (k1 + k4 + 2*(k2 + k3))/6;
```

e

```

===== F3.m =====
function yout = F3(t,p)
global alf bet ind
global a
global x
yout = a*x(ind)-alf*p;

```

Antes de iniciar a próxima parte, vamos conhecer primeiro alguns critérios de parada, além do proposto em E6 e em Armijo, também implementados para garantir a convergência dos métodos. São estes:

CP1: $\|\Psi_u(u^{(k)})\| < \varepsilon$.

CP2: $\|\Psi_u(u^{(k+1)}) - \Psi_u(u^{(k)})\| < \varepsilon$.

CP3: $\|u^{(k+1)} - u^{(k)}\| < \varepsilon$.

CP4: $\rho < \varepsilon_1$.

Como já obtemos o valor do gradiente $\Psi_u(u^{(k)})$, vamos iniciar pelo critério CP1.

```

===== PARTE 4 =====
normg = sqrt(sum(w.^2)) % calcula a norma do gradiente
ii = ii+1;
grad(ii) = normg;
% Verificando CP1
if normg < eps
    disp('CONVERGENCE by GRADIENT') % Conver. pelo Gradiente
    flag1 = 1;
    break
end

```

Se o critério é satisfeito, o indicador de convergência recebe o valor 1 e a afirmação *break* sai do loop do gradiente iniciado na PARTE 2. Continuando o processo, vamos para o passo E4 que requer o método de Armijo para o cálculo de ρ . Assim sendo, considere $\Psi(u) = \psi(u, x)$, Armijo para esse problema é formulado abaixo.

Método de Armijo para o passo ρ

A0: Seja $\bar{\rho} := \rho_{k-1}$.

A1: $u := u^{(k)} - \bar{\rho}\omega^{(k)}$.

A2: Calcule a solução x_1 do problema estado com entrada u :

$$\begin{cases} x_1'(t) = \alpha x(t) + \beta u(u), & t \in (0, T) \\ x_1(0) = x_0 \end{cases}$$

A3: Calcule $\psi(u, x_1)$.

A4: (O critério de parada)

Se $\psi(u, x_1) \geq \psi(u^{(k)}, x^{(k)})$

então $\bar{\rho} := b\bar{\rho}$; vá para A1

senão $\rho_k := \bar{\rho}$; PARAR algoritmo.

Então, a parte do programa referente a esse método é

```

===== PARTE 5 =====
% E4: Encontrando RO
robar = ro;
flag2 = 0; % indicador de convergência para o alg. de Armijo
flag3 = 0; % indic. de converg. quando o loop for finalizado por CP4
% Iniciando loop para encontrar rho
for count = 1:maxro
    count % variável que controla o loop
    u = uold - robar*w;
    % resolver a equação estado com entrada u e estado x1
    x1(1) = x0;
    for j = 1:n-1
        ind = j;
        x1(j+1) = RK42(x1(j),t(j));
    end
    % teste para CP4
    if robar < eps1
        flag3 = 1;
        break % sai do loop count
    end
    % calcular o valor do custo atual e testar rho
    Q1 = a*x1.^2 + u.^2;
    temp = trapz(t,Q1);
    cv = temp/2;
    if cv >= cvold % se não diminuir o custo para a minimização
        robar = bro * robar
    else
        flag2 = 1;
        break % sai do loop count
    end
end % fim do loop count

```

onde a função $RK42.m$ é similar a $RK41.m$, porém com $F2.m$ no lugar de $F1.m$, é dada por

```
===== F2.m =====
function yout = F2(t,x)
global alf bet ind
global u
yout = alf*x + bet*u(ind);
```

A PARTE 6 abaixo contém apenas o teste dos critérios de parada CP2 e CP3 para o método de Armijo poder encontrar o passo ρ_k .

```
===== PARTE 6 =====
if flag3 == 1
    disp('CONVERGENCE BY RO') % Convergência por RO
    break % sai do loop iter ... (CP4 satisfeito)
end
if flag2 == 1
    cvnew = cv
    jj = jj+1;
    cost(jj) = cvnew;
    unew = u;
    ro = robar;
    % Verificando CP2
    if abs(cvnew-cvold) < eps
        disp('CONVERGENCE by COST') % Convergência pelo Custo
        flag1 = 1;
        break % sai do loop iter ... (CP2 satisfeito)
    end
    % Verificando CP3
    d = uold-unew;
    dif = sqrt(sum(d.^2));
    if dif < eps
        disp('CONVERGENCE by CONTROL') % Conv. pelo controle
        flag1 = 1;
        break % sai do loop iter ... (CP3 satisfeito)
    end
else
    error('NO CONVERGENCE FOR RO') % PARAR PROGRAMA
end
```

Encerramos, então, o loop do algoritmo do gradiente mostrando os resultados e os gráficos para as variáveis ótimas caso a convergência seja obtida:

```

===== PARTE 7 =====
% preparar uma nova iteração
uold = unew;
cvold = cvnew;
%toma x = x1; para ser introduzido na versão final abaixo
end % finaliza o loop iter
if (flag1 == 1) || (flag3 == 1) % || significa 'ou'
    plot(t,unew,'*'); grid % exibe o gráfico da função unew
    xlabel('\bf t','FontSize',16)
    ylabel('\bf u(t)','FontSize',16)
    figure(2)
    plot(t,x1,'r*'); grid % exibe o gráfico da função x1
    xlabel('\bf t','FontSize',16)
    ylabel('\bf x(t)','FontSize',16)
else
    error('NO CONVERGENCE FOR DESCENT METHOD')
end
grad = grad';
cost = cost';
save grad.txt grad -ascii % salva o vetor grad no arquivo grad.txt
save cost.txt cost -ascii % salva o vetor cost no arquivo cost.txt
disp('END OF JOB')

```

O programa finaliza aqui. Após digitar o nome do programa, da maneira como foi salvo, na janela de comando do MATLAB, um teste numérico que pode ser feito é tomando os valores $T = 1, \alpha = 2, \beta = 0.7, a = 3, h = 0.001, r_0 = 1, maxro = 20, bro = 0.55, \varepsilon = \varepsilon_1 = 0.001, x_0 = 0, u_0 = 10$. O programa converge numericamente para o par ótimo e o valor do custo funcional, $1.0037e^{-16}$, é obtido após 53 iterações [(ANITA; ARNAUTU; CAPASSO, 2011), Seção 3.2].

Seguimos agora para a construção do programa para o método de varredura frente-trás, baseado em (LENHART; WORKMAN, 2007), Capítulo 4. Pelos passos S1 - S5 descritos na seção anterior, temos:

Método de Varredura Frente-Trás para o problema (P2)

```

===== ExeMFT.m =====
function y = ExeMFT(A,B,a,x0)
%cria a função ExeMFT com variáveis A, B, a, x0, onde A =  $\alpha$  e B =  $\beta$ 
test = -1; % variável do teste de convergência
delta = 0.001;
N = 1000;
t = linspace(0,1,N+1); % cria N+1 pontos igualm. espaçados entre 0 e 1
h = 1/N; % h é o espaçamento entre os pontos
h2 = h/2;
u = zeros(1,N+1); % estimativa inicial para o vetor u
x = zeros(1,N+1); % declara o vetor x seu tamanho
x(1) = x0; % valor inicial para o vetor x
lambda = zeros(1,N+1); % declara o vetor lambda e seu vetor
while(test < 0) % início do loop
    % armazena os vetores u, x e  $\lambda$  como valores anteriores
    oldu = u;
    oldx = x;
    oldlambda = lambda;
    % Runge-Kutta avançando no tempo para a variável estado  $x(t)$ 
    for i = 1:N
        k1 = A*x(i)+ B*u(i);
        k2 = A*(x(i)+h2*k1)+B*0.5*(u(i)+u(i+1));
        k3 = A*(x(i)+h2*k2)+B*0.5*(u(i)+u(i+1));
        k4 = A*(x(i)+h*k3)+B*u(i+1);
        x(i+1) = x(i) + (h/6)*(k1+2*k2+2*k3+k4);
    end
    % Runge-Kutta voltando no tempo para a variável adjunta  $\lambda(t)$ 
    for i = 1:N
        j = N+2-i;
        k1 = -A*lambda(j)+a*x(j);
        k2 = -A*(lambda(j)-h2*k1)+a*0.5*(x(j)+x(j-1));
        k3 = -A*(lambda(j)-h2*k2)+a*0.5*(x(j)+x(j-1));
        k4 = -A*(lambda(j)-h*k3)+a*x(j-1);
        lambda(j-1)=lambda(j)-(h/6)*(k1+2*k2+2*k3+k4);
    end
    end
    u1 = B*lambda; % representação de u com um novo valor para  $\vec{\lambda}$ 
    u = 0.5*(u1 + oldu);

```

```

===== ExeMFT.m =====
% parâmetros teste de convergência para cada variável com  $\delta = 0.001$ 
temp1 = delta*sum(abs(u)) - sum(abs(oldu-u));
temp2 = delta*sum(abs(x)) - sum(abs(oldx-x));
temp3 = delta*sum(abs(lambda)) - sum(abs(oldlambda-lambda));
test = min(temp1, min(temp2, temp3));
% a variável test obtém o menor dos três valores
end % fim do loop while
% Se converge, os vetores finais são armazenados na matriz de saída y
y(1, :) = t;
y(2, :) = x;
y(3, :) = lambda;
y(4, :) = u;

```

Ao contrário do método anterior, o algoritmo para Runge-Kutta não foi criado à parte, em um novo arquivo do MATLAB, mas, sim, implementado junto ao programa. Em suas linhas, o valor $k1$, no primeiro Runge-Kutta, corresponde à equação diferencial do estado. Em $k2$, x é substituído por $x + \frac{h}{2}k1$, a variável t por $t + \frac{h}{2}$ e para a outra variável presente, o controle, é necessário uma aproximação por meio da média

$$\frac{u(i) + u(i+1)}{2}.$$

Em $k3$ o procedimento é similar. Já em $k4$, utilizamos um passo inteiro no tempo, logo o próprio h e a variável $u(i+1)$ são aplicados. A última linha do algoritmo gera o próximo valor iterado do estado. O segundo algoritmo de Runge-Kutta, que calcula a variável adjunta voltando no tempo, é composto de maneira semelhante ao anterior, porém, como é esperado, fazendo uma subtração no tempo.

No MATLAB esse programa é executado após digitar na janela de comando o nome da função que o representa acompanhada dos valores de entrada. Por exemplo, aplicando os mesmos valores do método do gradiente:

```
ExeMFT(2,0.7,3,0);
```

obtemos os gráficos de $x(t)$ e $u(t)$ com respeito ao tempo t digitando

```
plot((y(1,:), y(2,:)); e plot((y(1,:), y(4,:));
```

respectivamente.

Dessa forma, temos uma base de como esses métodos são introduzidos e quais ferramentas utilizamos para melhor descrevermos.

3 Controle Ótimo Aplicado à Dengue

Devido a numerosos surtos de doenças virais, como dengue e chikungunya, nos últimos tempos, vários são os tipos de controle estudados para minimizar alguns fatores presentes. Como por exemplo, o número de indivíduos infectados pelo vírus causador da doença. Será apresentado neste capítulo um problema de controle ótimo baseado em (RODRIGUES; MONTEIRO; TORRES, 2014) cujo intuito é minimizar a quantia de infectados.

Assim sendo, considere o modelo epidemiológico proposto por (RODRIGUES; MONTEIRO; TORRES, 2014) e formado por cinco variáveis estado dentre as quais três correspondem a estados epidemiológicos para os humanos:

$$\begin{aligned} S_h(t) &= \text{indivíduos suscetíveis (que podem contrair a doença);} \\ I_h(t) &= \text{indivíduos infectados (que podem transmitir a doença);} \\ R_h(t) &= \text{indivíduos resistentes (que foram infectados e se recuperaram).} \end{aligned}$$

E as outras duas se referem aos mosquitos (exclusivamente fêmeas já que os machos não se alimentam de sangue):

$$\begin{aligned} S_m(t) &= \text{mosquitos suscetíveis (que podem contrair a doença);} \\ I_m(t) &= \text{mosquitos infectados (que podem transmitir a doença).} \end{aligned}$$

Para esse modelo epidemiológico algumas afirmações devem ser consideradas, como:

- Não existe transmissão vertical, ou seja, o mosquito infectado não pode transmitir a doença aos seus ovos;
- A população humana N_h é constante: $S_h(t) + I_h(t) + R_h(t) = N_h$ em qualquer tempo t ;
- A população de mosquito também é constante e proporcional à humana, isto é, $S_m(t) + I_m(t) = N_m$ com $N_m = kN_h$ para alguma constante k ;
- A população é homogênea, ou seja, cada indivíduo classificado em um estado acima é homogeneamente misturado com os outros indivíduos;
- Imigração e emigração não são consideradas durante o período estudado;
- Homogeneidade entre as populações de hospedeiro e vetor, isto é, cada vetor tem igual probabilidade de infectar qualquer hospedeiro;
- Assumimos que humanos e mosquitos já nascem suscetíveis.

Dessa forma, o sistema diferencial formado pelas variáveis estados para o problema ótimo é:

$$\left\{ \begin{array}{l} \frac{dS_h(t)}{dt} = \mu_h N_h - \left(B\beta_{mh} \frac{I_m(t)}{N_h} + \mu_h \right) S_h(t) \\ \frac{dI_h(t)}{dt} = B\beta_{mh} \frac{I_m(t)}{N_h} S_h(t) - (\eta_h + \mu_h) I_h(t) \\ \frac{dR_h(t)}{dt} = \eta_h I_h(t) - \mu_h R_h(t) \\ \frac{dS_m(t)}{dt} = \mu_m N_m - \left(B\beta_{hm} \frac{I_h(t)}{N_h} + \mu_m + u(t) \right) S_m(t) \\ \frac{dI_m(t)}{dt} = B\beta_{hm} \frac{I_h(t)}{N_h} S_m(t) - (\mu_m + u(t)) I_m(t). \end{array} \right. \quad (3.0.1)$$

Com condições iniciais

$$\begin{aligned} S_h(0) &= S_{h0}, & I_h(0) &= I_{h0}, & R_h(0) &= R_{h0}, \\ S_m(0) &= S_{m0}, & I_m(0) &= I_{m0}. \end{aligned} \quad (3.0.2)$$

Onde N_h e N_m são a população total humana e a população total de mosquito, respectivamente; B é a média diária de picada dos mosquitos; β_{mh} e β_{hm} correspondem à probabilidade de transmissão dos mosquitos para os humanos e dos humanos para os mosquitos, respectivamente, isto é, a probabilidade de um mosquito infeccioso transmitir a doença a um humano suscetível e vice-versa. Assim, assumindo que BN_m/N_h é o número médio de picadas que um humano recebe, podemos dizer que os quocientes abaixo, presentes no sistema (3.0.1), se referem às taxas de infecção por humano suscetível e por mosquito suscetível, respectivamente:

$$B\beta_{mh} \frac{N_m}{N_h} \frac{I_m(t)}{N_m} = B\beta_{mh} \frac{I_m(t)}{N_h} \quad \text{e} \quad B\beta_{hm} \frac{I_h(t)}{N_h}.$$

No mais, η_h é a taxa de recuperação dos humanos, com $1/\eta_h$ sendo o período médio virêmico em dias, ou seja, o tempo médio em que a doença fica presente no organismo; após o qual o ser humano (antes infectado) se torna resistente; μ_h e μ_m são as expectativas de vida dos humanos e mosquitos em dias, que neste modelo foram consideradas 79 anos e 10,95 dias, respectivamente, e, com período médio de vida para ambos representado por $1/\mu_h$ e $1/\mu_m$. A tabela 1 resume estes parâmetros e a Figura 1 mostra um esquema para obter as equações do modelo.

Tabela 1: Parâmetros do sistema (3.0.1).

Parâmetros	Descrição
N_h	população total humana
N_m	população total de mosquito
B	média diária de picada
β_{mh}	probabilidade de transmissão pelo I_m (por picada)
β_{hm}	probabilidade de transmissão pelo I_h (por picada)
$1/\mu_h$	tempo médio de vida dos humanos (em dias)
$1/\mu_m$	tempo médio de vida dos mosquitos (em dias)
$1/\eta_h$	período virêmico médio (em dias)

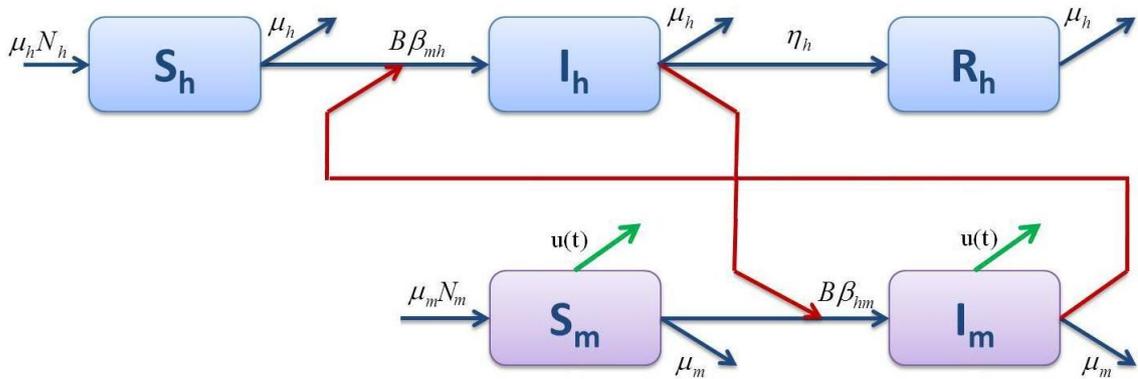


Figura 1: Modelo epidemiológico SIR. Fonte: (RODRIGUES et al., 2014)

Podemos observar que um controle $u(t)$ foi introduzido no sistema referente aos mosquitos; $u(t)$ corresponde ao controle inseticida. O objetivo, então, é minimizar o número de indivíduos infectados I_h , mantendo baixo o custo da estratégia de implementação do controle, ou seja,

$$\min_{u \in \Omega} \phi(u) = \int_0^T [\gamma_D I_h(t) + \gamma_S u^2(t)] dt \quad (P3)$$

onde γ_D e γ_S representam, respectivamente, os fatores do custo de equilíbrio para indivíduos infectados e as campanhas de pulverização. O conjunto Ω dos possíveis controles é dado por $\Omega = \{u \in L^1(0, T) | 0 \leq u(t) \leq 1\}$, isto significa que os controles obtidos estarão limitados inferiormente e superiormente. Toda a teoria vista no Capítulo 1 é válida e pode ser estendida para problemas com controle limitado [ver (LENHART; WORKMAN, 2007), Capítulo 8]. Assim, a existência do controle ótimo $u^*(\cdot)$ é garantida pela convexidade do objetivo funcional em (P3). Pelo Princípio de Pontryagin, se $u^*(\cdot)$ é ótimo para o problema (P3) com equações estado (3.0.1) e condições iniciais (3.0.2) no intervalo $[0, T]$, então existe um vetor adjunto

$$\vec{\lambda}(t) = (\lambda_1(t), \lambda_2(t), \lambda_3(t), \lambda_4(t), \lambda_5(t))$$

tal que, renomeando as variáveis estado

$$\begin{aligned}\vec{x}(t) &= (x_1(t), x_2(t), x_3(t), x_4(t), x_5(t)) = (S_h(t), I_h(t), R_h(t), S_m(t), I_m(t)) \\ \vec{x}_0 &= (x_{10}, x_{20}, x_{30}, x_{40}, x_{50}) = (S_{h0}, I_{h0}, R_{h0}, S_{m0}, I_{m0}),\end{aligned}$$

temos

$$\dot{x}_i = \frac{\partial H}{\partial \lambda_i} \quad \text{e} \quad \dot{\lambda}_i = -\frac{\partial H}{\partial x_i}, \quad \forall i = 1, \dots, 5,$$

onde o Hamiltoniano H é definido por

$$\begin{aligned}H(t, u, \vec{x}, \vec{\lambda}) &= \gamma_D x_2 + \gamma_s u(t)^2 + \lambda_1 \left(\mu_h N_h - \left(B\beta_{mh} \frac{x_5}{N_h} + \mu_h \right) x_1 \right) \\ &+ \lambda_2 \left(B\beta_{mh} \frac{x_5}{N_h} x_1 - (\eta_h + \mu_h) x_2 \right) \\ &+ \lambda_3 (\eta_h x_2 - \mu_h x_3) \\ &+ \lambda_4 \left(\mu_m N_m - \left(B\beta_{hm} \frac{x_2}{N_h} + \mu_m + u(t) \right) x_4 \right) \\ &+ \lambda_5 \left(B\beta_{hm} \frac{x_2}{N_h} x_4 - (\mu_m + u(t)) x_5 \right).\end{aligned}$$

Como o problema possui cinco equações estado, então é necessário uma mesma quantidade de variáveis adjuntas. Este problema é característico de controle ótimo com várias variáveis. Uma forma geral desse tipo de controle é dada no Apêndice B [ou para mais informações veja (LENHART; WORKMAN, 2007)].

As funções adjuntas do Hamiltoniano acima satisfazem

$$\left\{ \begin{array}{l} \dot{\lambda}_1 = (\lambda_1(t) - \lambda_2(t)) B\beta_{mh} \frac{x_5(t)}{N_h} + \lambda_1(t) \mu_h \\ \dot{\lambda}_2 = (\lambda_4(t) - \lambda_5(t)) B\beta_{hm} \frac{x_4(t)}{N_h} + \lambda_2(t) (\eta_h + \mu_h) - (\lambda_3(t) \eta_h + \gamma_D) \\ \dot{\lambda}_3 = \lambda_3(t) \mu_h \\ \dot{\lambda}_4 = (\lambda_4(t) - \lambda_5(t)) B\beta_{hm} \frac{x_2(t)}{N_h} + \lambda_4(t) (\mu_m + u(t)) \\ \dot{\lambda}_5 = (\lambda_1(t) - \lambda_2(t)) B\beta_{mh} \frac{x_1(t)}{N_h} + \lambda_5(t) (\mu_m + u(t)), \end{array} \right. \quad (3.0.3)$$

com condições de transversalidade $\lambda_i(T) = 0$ para todo $i = 1, \dots, 5$. Já que $u(t)$ é limitado, ele é escrito da seguinte forma no programa:

$$u^*(t) = \min \left\{ \max \left\{ 0, \frac{\lambda_4(t) x_4^*(t) + \lambda_5 x_5^*(t)}{2\gamma_s} \right\}, 1 \right\}.$$

O problema ótimo primeiramente foi resolvido no MATLAB através do Método de Varredura Frente-Trás, abordado no Capítulo 1. Os valores dos parâmetros foram também baseados em (RODRIGUES; MONTEIRO; TORRES, 2014) que correspondem a

um cenário cuja temperatura média de 26°C. Como o modelo epidemiológico é influenciado pela temperatura, já que o aumento desta facilita a disseminação do mosquito e do vírus presente nele, um cenário desse tipo é bastante estudado por ser o mais ameaçador à saúde pública. Os valores dos parâmetros e das condições iniciais estão descritos na Tabela 2.

Tabela 2: Valores dos parâmetros e condições iniciais do problema (P3). Fonte: (RODRIGUES; MONTEIRO; TORRES, 2014)

Parâmetros	Valores	Cond. Iniciais	Valores
N_h	112000	x_{10}	111991
N_m	336000	x_{20}	9
B	1/3	x_{30}	0
β_{mh}	0.99	x_{40}	335000
β_{hm}	0.95	x_{50}	1000
$1/\mu_h$	1/(79x365)		
$1/\eta_h$	1/7		
$1/\mu_m$	0.03		

O programa escrito no MATLAB para resolver este problema se encontra no Apêndice B.2. As Figuras 2, 3 e 4 foram resultadas do programa implementado.

Observando a Figura 2, podemos notar que a implementação de um controle no problema diminui drasticamente o número de indivíduos infectados. Em uma população total de 112000 pessoas, o número de infectados chegou ao seu máximo em apenas 10 dias com uma quantia de 264 infectados, que corresponde a 0,25% da população total, e logo após foi reduzido a zero após 183 dias, permanecendo assim até o fim do período de 365 dias.

Já na estratégia sem controle, Figura 3, a doença causou um número de 56000 infectados; 50% da população; não chegando a zerar esse número, ou seja, ao fim dos 365 dias ainda havia 24 infectados. Após o décimo dia, na presença de controle, enquanto os infectados já começavam a diminuir, na ausência de controle estes chegaram a uma quantia de 914 pessoas e continuou a crescer até os 60 primeiros dias, só diminuindo após esse tempo.

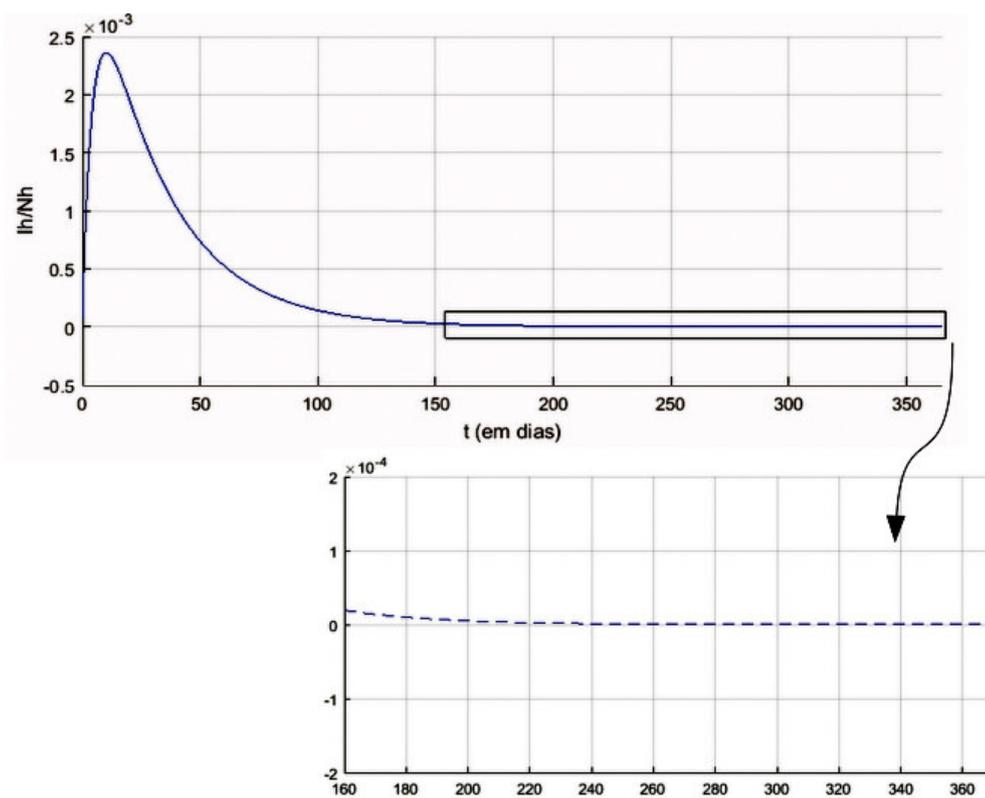


Figura 2: Fração do número de infectados I_h/N_h na presença de controle.

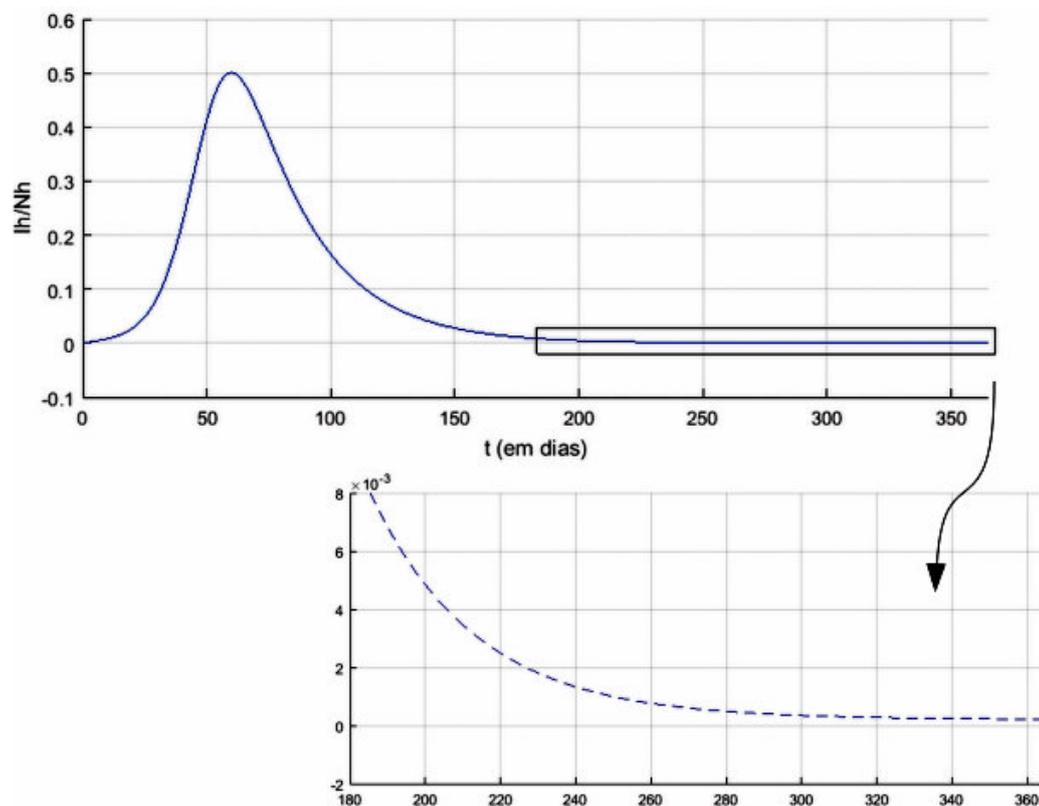


Figura 3: Fração do número de infectados I_h/N_h na ausência de controle.

Na Figura 4, vemos que para reduzir a quantidade de infeccioso, o controle ótimo $u^*(t)$ iniciou próximo ao seu limite superior e se manteve até 324 dias, quando começou a se aproximar do seu limite inferior, já que o número de infectados era praticamente nulo.

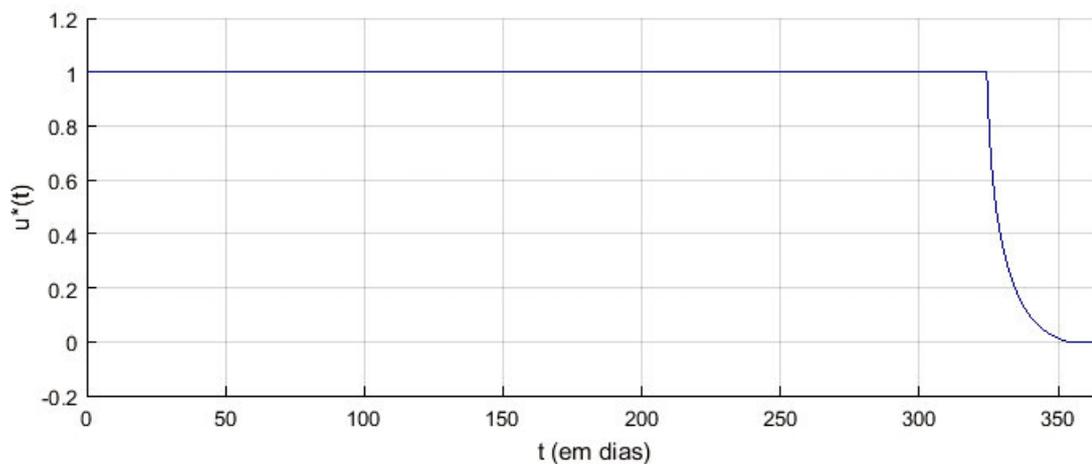


Figura 4: Controle Ótimo $u^*(t)$ para a Figura 2.

Para estes resultados, os fatores econômico e social presentes no objetivo funcional foram considerados como sendo $\gamma_D = \gamma_S = 1$. A convergência do programa para o cenário da Figura 2 foi obtida após 21 iterações enquanto que a do cenário da Figura 3 foi obtida com 2 iterações.

4 Considerações Finais

Como mencionado na Introdução, diversas análises podem ser realizadas diante de problemas epidemiológicos, alterando parâmetros, acrescentando controles ou outros estados ao modelo, introduzindo elementos ambientais que também alteram o comportamento dos parâmetros, como a temperatura, a pluviosidade; dentre outras análises.

No caso trabalhado, pudemos ver que o uso de um controle foi imprescindível para se chegar ao objetivo em questão de reduzir o número de infectados pelo vírus da dengue em período de 365 dias a uma temperatura ambiente de 26° C. Ou seja, em apenas um modelo SIR simples já compreendemos a importância das técnicas de controle, o que nos dá uma base para lidarmos com problemas epidemiológicos mais complexos.

Atualmente, além do objetivo a ser alcançado nessas modelagens epidemiológicas, o controle ótimo também é essencial para associar os fatores econômicos dos modelos. Portanto, com uma alteração no custo funcional, obtemos resultados que envolvem tanto a perspectiva humana quanto a econômica.

Referências

- ANITA, S.; ARNAUTU, V.; CAPASSO, V. *An Introduction to Optimal Control Problems in Life Sciences and Economics. From Mathematical Models to Numerical Simulation with MATLAB*. [S.l.]: Birkhauser, New York, 2011. Citado 10 vezes nas páginas 11, 13, 15, 17, 20, 21, 22, 23, 30 e 45.
- ARMIJO, L. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, H. Samelson; R. M. Blumenthal; J. Dugundji; Richard Arens, v. 16, n. 1, p. 1–3, 1966. Citado na página 21.
- ARNAUTU, V.; NEITTAANMAKI, P. *Optimal Control from Theory to Computer Programs*. [S.l.]: Kluwer Academic, Dordrecht, 2003. Citado 2 vezes nas páginas 17 e 21.
- BAUER, J. *Implementação de Algoritmos de Programação Não Linear*. TCC, 2004. Citado na página 20.
- BUTCHER, J. On the convergence of numerical solutions of ordinary differential equations. *Mathematics of Computation*, American Mathematical Society, v. 20, n. 93, p. 1–10, 1966. Citado na página 46.
- BUTCHER, J. A multistep generalization of runge-kutta methods with four or five stages. *Journal of the Association for Computing Machinery*, ACM New York, v. 14, n. 93, p. 84–99, 1967. Citado na página 46.
- ISERLES, A. *A First Course in the Numerical Analysis of Differential Equations*. [S.l.]: Cambridge University Press, New York, 1996. Citado na página 46.
- IZMAILOV A.; SOLODOV, M. *Otimização - Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*. [S.l.]: IMPA, Títulos de Otimização, Rio de Janeiro, 2014. v. 1. Citado na página 11.
- LENHART, S.; WORKMAN, J. T. *Optimal Control Applied to Biological Models. Chapman & Hall/CRC Mathematical and Computational Biology Series*. [S.l.]: Boca Raton, FL, 2007. Citado 11 vezes nas páginas 11, 13, 14, 15, 18, 19, 30, 35, 36, 44 e 47.
- RODRIGUES, H. S.; MONTEIRO, M. T. T.; TORRES, D. F. M. Seasonality effects on dengue: basic reproduction number, sensitivity analysis and optimal control. *Mathematical Methods in the Applied Sciences*, arXiv: 1409.3928v1, p. 1–13, 2014. Citado 5 vezes nas páginas 9, 12, 33, 36 e 37.
- RODRIGUES, H. S. et al. Dengue in madeira island. *Mathematics of Planet Earth, CIM Mathematical Sciences, Springer*, p. 1–13, 2014. Citado 2 vezes nas páginas 8 e 35.
- SAÚDE, M. da. *Prevenção e Combate: Dengue, Chikungunya e Zika*. Disponível em: <<http://combateaedes.saude.gov.br/pt/tira-duvidas>>. Citado na página 12.
- SAÚDE, P. da. *Orientações Gerais, Prevenção e Combate: Dengue, Chikungunya e Zika*. Disponível em: <<http://portalsaude.saude.gov.br/index.php/o-ministerio/principal/secretarias/svs/dengue>>. Citado na página 12.

Apêndices

APÊNDICE A –

A.1 O Espaço $L^2(\Omega)$

O espaço $L^2(\Omega)$ é um caso particular dos espaços $L^p(\Omega)$ de todas as funções mensuráveis para $p = 2$ com Ω um subconjunto mensurável limitado. $L^2(\Omega)$ é um espaço de Hilbert munido com o produto interno

$$(f, g)_{L^2(\Omega)} = \int_{\Omega} f(x)g(x)dx.$$

Com $f, g : \Omega \rightarrow \mathbb{R}$.

A.2 Definições

Seja V um espaço de Hilbert (espaço vetorial com produto interno canônico).

Definição A.1. $K \subseteq V$ é um **conjunto convexo** se $\lambda x_1 + (1 - \lambda)x_2 \in K$ para qualquer $x_1, x_2 \in K$ e qualquer $\lambda \in [0, 1]$.

Definição A.2. $f : V \rightarrow \mathbb{R}$ é uma **função convexa** se

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

para qualquer $x_1, x_2 \in K$ e qualquer $\lambda \in [0, 1]$.

Definição A.3. Seja $I \subseteq \mathbb{R}$ um intervalo. Uma função $f : I \rightarrow V$ é **contínua por partes** se ela é contínua em todo $t \in I$, exceto numa quantidade finita de t , e se ela é igual a seu limite esquerdo ou ao seu direito em todo $t \in I$.

Definição A.4. Seja $g : I \rightarrow V$ contínua em $I \subseteq \mathbb{R}$ e diferenciável em todos os seus pontos, exceto numa quantidade finita. Além disso, suponha que g' é contínua em todo seu domínio. Então, g é **diferenciável por partes**.

Definição A.5. Considere a função $k : I \rightarrow V$. Dizemos que k é **continuamente diferenciável** se k' existe e é contínua em $I \subseteq \mathbb{R}$.

Definição A.6. Considere os espaços normados V e H e o operador $F : D \subseteq V \rightarrow H$. A **derivada direcional** de F em $x \in D$ ao longo da direção $h \in V$ é definida por

$$F'(x, h) = \lim_{\lambda \rightarrow 0} \frac{F(x + \lambda h) - F(x)}{\lambda},$$

se o limite existir.

Definição A.7. Se para todo $x \in D$ a derivada $F'(x, h)$ existe ao longo de qualquer direção h e se o operador $h \mapsto F'(x, h)$ é linear e contínuo, então F é **Gâteaux diferenciável** em x . Nesse caso, $F'(x) \in L(V, H)$ definida por

$$F'(x)h = \lim_{\lambda \rightarrow 0} \frac{F(x + \lambda h) - F(x)}{\lambda},$$

é a Gâteaux derivada de F em x , também denotada por $F_x(x)$ ou $dF(x)$.

A.3 Interpretação da Adjunta

Para entendermos a variável adjunta, definimos uma função V por

$$V(x_0, t_0) := \max_u \int_{t_0}^{t_1} f(t, u(t), x(t)) dt$$

$$\text{sujeito a } x'(t) = g(t, u(t), x(t)); \quad x(t_0) = x_0.$$

A função V corresponde ao valor da integral quando avaliada no estado e controle ótimos. A notação $V(x_0, t_0)$ indica a dependência sobre o estado e o tempo iniciais. Há uma relação entre a variável adjunta e a derivada de V com respeito ao estado, isto é,

$$\frac{\partial V}{\partial x}(x_0, t_0) = \lim_{\varepsilon \rightarrow 0} \frac{V(x_0 + \varepsilon, t_0) - V(x_0, t_0)}{\varepsilon} = \lambda(t_0).$$

Assim, temos que a variável adjunta $\lambda(t_0)$ é igual a uma variação na função com respeito ao estado no tempo inicial t_0 . Por exemplo, em um problema de pescaria onde o estado representa a quantidade de peixes e o objetivo funcional representa o lucro obtido na pescaria, a variável adjunta pode ser interpretada como: "*Se um peixe é adicionado ao estoque no tempo t_0 , quanto o valor da pescaria é afetado?*". De fato, seguindo nessa interpretação, podemos generalizá-la para todo tempo t :

$$\frac{\partial V}{\partial x}(x^*(t), t) = \lambda(t) \quad \forall t_0 \leq t \leq t_1.$$

[(LENHART; WORKMAN, 2007), Seção 2.2].

A.4 O Cone Normal

Seja K um subconjunto convexo fechado do espaço de Hilbert V munido com um produto interno (\cdot, \cdot) .

Definição A.8. $K \subseteq V$ é um **cone** se $\lambda x \in K$ para todo $x \in K$ e $\lambda > 0$.

Definição A.9. Dado $u \in K$, o conjunto

$$N_K(u) = \{w \in V; (v - u, w) \leq 0 \text{ para qualquer } v \in K\}$$

é chamado o **cone normal** de K em u .

Exemplo A.1. Se $V = \mathbb{R}$, $K = [a, b]$ ($a, b \in \mathbb{R}, a < b$), então

$$N_K(u) = \begin{cases} \mathbb{R}^+, & u = b \\ \{0\}, & a < u < b \\ \mathbb{R}^-, & u = a. \end{cases}$$

Proposição A.1. *Sejam $\Omega \subset \mathbb{R}^m$ ($m \in \mathbb{N}^*$) um subconjunto aberto e limitado, e $\omega_1, \omega_2 \in L^2(\Omega)$, com $\omega_1(x) < \omega_2(x)$ em quase todo ponto $x \in \Omega$. Defina*

$$K = \{y \in L^2(\Omega); \omega_1(x) \leq y(x) \leq \omega_2(x) \text{ q.t.p. } x \in \Omega\}.$$

K é o subconjunto convexo fechado de $L^2(\Omega)$. Então para qualquer $z \in N_K(y)$ temos

$$\begin{aligned} z(x) &\leq 0 \text{ para } y(x) = \omega_1(x); \\ z(x) &\geq 0 \text{ para } y(x) = \omega_2(x); \\ z(x) &= 0 \text{ para } \omega_1(x) < y(x) < \omega_2(x), \end{aligned}$$

quase todo ponto em Ω .

Prova:

Considere $A = \{x \in \Omega; y(x) = \omega_1(x)\}$. Vamos provar que $z(x) \leq 0$ q.t.p. $x \in A$. Por contradição, assuma que existe um subconjunto $\tilde{A} \subset A$ tal que $\mu(\tilde{A}) > 0$ e $z(x) > 0$ q.t.p. $x \in \tilde{A}$. Como $z \in N_K(y)$, temos que

$$\int_{\Omega} z(x)(h(x) - y(x))dx \leq 0,$$

para qualquer $h \in K$. Então, escolhemos $h \in K$ tal que

$$\begin{aligned} h(x) &= \omega_1(x) \text{ q.t.p. } x \in \Omega \setminus \tilde{A}, \\ h(x) &> \omega_1(x) \text{ q.t.p. } x \in \tilde{A}. \end{aligned}$$

Assim,

$$\int_{\Omega} z(x)(h(x) - y(x))dx = \int_{\tilde{A}} z(x)(h(x) - y(x))dx > 0,$$

que é um absurdo. Logo, $z(x) \leq 0$ q.t.p. $x \in A$. Da mesma maneira obtemos os outros dois pontos. \square

Para uma melhor compreensão de alguns conceitos presentes na prova da proposição acima, consulte (ANITA; ARNAUTU; CAPASSO, 2011), Apêndices A.1.1 e A.1.2.

A.5 Runge-Kutta de Quarta Ordem (RK4)

Considere um passo de tamanho h e uma EDO

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0$$

a solução aproximada $y(t + h)$ da equação acima por Runge-Kutta de quarta ordem é dada por

$$y(t + h) = y(t) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

onde

$$\begin{aligned}k_1 &= f(t, y(t)) \\k_2 &= f\left(t + \frac{h}{2}, y(t) + \frac{h}{2}k_1\right) \\k_3 &= f\left(t + \frac{h}{2}, y(t) + \frac{h}{2}k_2\right) \\k_4 &= f(t + h, y(t) + hk_3).\end{aligned}$$

Isto é, a aproximação $y(t + h)$ é determinada pelo valor de $y(t)$ somado com o produto entre o passo h e a média ponderada das inclinações k_1, k_2, k_3 e k_4 em certos pontos do intervalo. Estas equações são obtidas por uma comparação com o polinômio de Taylor truncado na quarta ordem. Mais informações sobre como obter as inclinações, estabilidade e convergência de Runge-Kutta encontramos em (BUTCHER, 1966), (BUTCHER, 1967) e (ISERLES, 1996).

APÊNDICE B –

B.1 Controle Ótimo de Várias Variáveis

Considere o seguinte problema ótimo com n variáveis estado e controle dispondo das mesmas condições do problema (P1):

$$\max_{\vec{u}} \int_0^T f(t, \vec{u}(t), \vec{x}(t)) dt$$

sujeito a

$$\vec{x}'(t) = \vec{g}(t, \vec{u}(t), \vec{x}(t)) \quad \vec{x}(t_0) = \vec{x}_0.$$

Onde $\vec{u}, \vec{x}, \vec{x}_0, \vec{g}$ são vetores da forma

$$\vec{u}(t) = (u_1(t), u_2(t), \dots, u_n(t))$$

$$\vec{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$$

$$\vec{x}_0 = (x_{10}, x_{20}, \dots, x_{n0})$$

$$\vec{g}(t, \vec{u}(t), \vec{x}(t)) = (g_1(t, \vec{u}(t), \vec{x}(t)), g_2(t, \vec{u}(t), \vec{x}(t)), \dots, g_n(t, \vec{u}(t), \vec{x}(t))),$$

com $n > 1, T > 0$. Assim, se (\vec{u}^*, \vec{x}^*) é um par ótimo para o problema acima, pelo Teorema 1.1 (Pontryagin), existe uma variável adjunta diferenciável por partes, também vetorial, da forma $\vec{\lambda}(t) = (\lambda_1(t), \dots, \lambda_n(t))$, onde cada λ_i corresponde a x_i . E, o Hamiltoniano pode ser definido como

$$H(t, \vec{u}(t), \vec{x}(t), \vec{\lambda}) = f(t, \vec{u}(t), \vec{x}(t)) + \vec{\lambda} \cdot \vec{g}(t, \vec{u}(t), \vec{x}(t)),$$

com \cdot representando um produto vetorial. Seguindo os mesmos argumentos do Capítulo 1, $\vec{u}, \vec{x}, \vec{\lambda}$ satisfazem

$$x'_i(t) = \frac{\partial H}{\partial \lambda_i} = g_i(t, \vec{u}(t), \vec{x}(t)), x_i(t_0) = x_{i0} \text{ para } i = 1, \dots, n,$$

$$\lambda'_i(t) = -\frac{\partial H}{\partial x_j} = g_j(t, \vec{u}(t), \vec{x}(t)), \lambda_j(T) = 0 \text{ para todo } j = 1, \dots, n,$$

$$0 = \frac{\partial H}{\partial u_k} \text{ para } k = 1, \dots, n.$$

Para mais informações sobre controle ótimo com várias variáveis veja (LENHART; WORKMAN, 2007) Capítulo 12.

B.2 Programa desenvolvido para o problema P3

Método de Varredura Frente-Trás para o problema P3:

```
===== dengue.m =====  
function y = dengue(MIh,Nh,B,Bmh,ETAh,MIm,Nm,Bhm,GD,GS,x10,x20,x30,x40,x50)  
  
test = -1;  
delta = 0.001;  
N = 1000;  
t = linspace(0,32000,N+1);  
h = 0.1;  
h2 = h/2;  
  
u = zeros(1,N+1);  
  
x1 = zeros(1,N+1);  
x2 = zeros(1,N+1);  
x3 = zeros(1,N+1);  
x4 = zeros(1,N+1);  
x5 = zeros(1,N+1);  
  
x1(1) = x10;  
x2(1) = x20;  
x3(1) = x30;  
x4(1) = x40;  
x5(1) = x50;  
  
% p = lambda  
p1 = zeros(1,N+1);  
p2 = zeros(1,N+1);  
p3 = zeros(1,N+1);  
p4 = zeros(1,N+1);  
p5 = zeros(1,N+1);  
  
while(test < 0)  
  
    oldu = u;
```

===== *dengue.m* =====

```
oldx1 = x1;
oldx2 = x2;
oldx3 = x3;
oldx4 = x4;
oldx5 = x5;
```

```
oldp1 = p1;
oldp2 = p2;
oldp3 = p3;
oldp4 = p4;
oldp5 = p5;
```

```
for i = 1:N
```

```
  k11 = MIh*Nh - (B*Bmh*(x5(i)/Nh)+MIh)*x1(i);
  k12 = B*Bmh*(x5(i)/Nh)*x1(i) - (ETAh + MIh)*x2(i);
  k13 = ETAh*x2(i)-MIh*x3(i);
  k14 = MIm*Nm-(B*Bhm*(x2(i)/Nh)+MIm+u(i))*x4(i);
  k15 = B*Bhm*(x2(i)/Nh)*x4(i) - (MIm+u(i))*x5(i);
```

```
  k21 = MIh*Nh - (B*Bmh*((x5(i)+h2*k15)/Nh)+MIh)*(x1(i)+h2*k11);
  k22 = B*Bmh*((x5(i)+h2*k15)/Nh)*(x1(i)+h2*k11)-(ETAh + MIh)*...
    (x2(i)+h2*k12);
  k23 = ETAh*(x2(i)+h2*k12)-MIh*(x3(i)+h2*k13);
  k24 = MIm*Nm-(B*Bhm*((x2(i)+h2*k12)/Nh)+MIm+...
    (0.5*(u(i)+u(i+1))))*(x4(i)+h2*k14);
  k25 = B*Bhm*((x2(i)+h2*k12)/Nh)*(x4(i)+h2*k14)-(MIm+...
    (0.5*(u(i)+u(i+1))))*(x5(i)+h2*k15);
```

```
  k31 = MIh*Nh - (B*Bmh*((x5(i)+h2*k25)/Nh)+MIh)*(x1(i)+h2*k21);
  k32 = B*Bmh*((x5(i)+h2*k25)/Nh)*(x1(i)+h2*k21)-(ETAh + MIh)*...
    (x2(i)+h2*k22);
  k33 = ETAh*(x2(i)+h2*k22)-MIh*(x3(i)+h2*k23);
  k34 = MIm*Nm-(B*Bhm*((x2(i)+h2*k22)/Nh)+MIm+(0.5*(u(i)+u(i+1))))*...
    (x4(i)+h2*k24);
  k35 = B*Bhm*((x2(i)+h2*k22)/Nh)*(x4(i)+h2*k24)-(MIm+...
    (0.5*(u(i)+u(i+1))))*(x5(i)+h2*k25);
```

```

===== dengue.m =====
k41 = MIh*Nh - (B*Bmh*((x5(i)+h*k35)/Nh)+MIh)*(x1(i)+h*k31);
k42 = B*Bmh*((x5(i)+h*k35)/Nh)*(x1(i)+h*k31)-(ETAh + MIh)*(x2(i)+h*k32);
k43 = ETAh*(x2(i)+h*k32)-MIh*(x3(i)+h*k33);
k44 = MIm*Nm-(B*Bhm*((x2(i)+h*k32)/Nh)+MIm+u(i+1))*(x4(i)+h*k34);
k45 = B*Bhm*((x2(i)+h*k32)/Nh)*(x4(i)+h*k34)-(MIm+u(i+1))*(x5(i)+h*k35);

x1(i+1) = x1(i) + (h/6)*(k11+2*k21+2*k31+k41);
x2(i+1) = x2(i) + (h/6)*(k12+2*k22+2*k32+k42);
x3(i+1) = x3(i) + (h/6)*(k13+2*k23+2*k33+k43);
x4(i+1) = x4(i) + (h/6)*(k14+2*k24+2*k34+k44);
x5(i+1) = x5(i) + (h/6)*(k15+2*k25+2*k35+k45);
end

for i = 1:N
  j = N + 2 - i;
  k11 = (p2(j)-p1(j))*B*Bmh*(x5(j)/Nh)-p1(j)*MIh;
  k12 = (p5(j)-p4(j))*B*Bhm*(x4(j)/Nh)-p2(j)*(ETAh+MIh)+(p3(j)*ETAh-GD);
  k13 = -p3(j)*MIh;
  k14 = (p5(j)-p4(j))*B*Bhm*(x2(j)/Nh)-p4(j)*(MIm+u(j));
  k15 = (p2(j)-p1(j))*B*Bmh*(x1(j)/Nh)-p5(j)*(MIm+u(j));

  k21 = ((p2(j)-h2*k12)-(p1(j)-h2*k11))*B*Bmh*(0.5*(x5(j)+x5(j-1))/Nh)-...
        (p1(j)-h2*k11)*MIh;
  k22 = ((p5(j)-h2*k15)-(p4(j)-h2*k14))*B*Bhm*(0.5*(x4(j)+x4(j-1))/Nh)-...
        (p2(j)-h2*k12)*(ETAh+MIh)+((p3(j)-h2*k13)*ETAh-GD);
  k23 = -(p3(j)-h2*k13)*MIh;
  k24 = ((p5(j)-h2*k15)-(p4(j)-h2*k14))*B*Bhm*(0.5*(x2(j)+x2(j-1))/Nh)-...
        (p4(j)-h2*k14)*(MIm+(0.5*(u(j)+u(j-1))));
  k25 = ((p2(j)-h2*k12)-(p1(j)-h2*k11))*B*Bmh*(0.5*(x1(j)+x1(j-1))/Nh)-...
        (p5(j)-h2*k15)*(MIm+(0.5*(u(j)+u(j-1))));

  k31 = ((p2(j)-h2*k22)-(p1(j)-h2*k21))*B*Bmh*(0.5*(x5(j)+x5(j-1))/Nh)-...
        (p1(j)-h2*k21)*MIh;
  k32 = ((p5(j)-h2*k25)-(p4(j)-h2*k24))*B*Bhm*(0.5*(x4(j)+x4(j-1))/Nh)-...
        (p2(j)-h2*k22)*(ETAh+MIh)+((p3(j)-h2*k23)*ETAh-GD);
  k33 = -(p3(j)-h2*k23)*MIh;

```

```

===== dengue.m =====
k34 = ((p5(j)-h2*k25)-(p4(j)-h2*k24))*B*Bhm*(0.5*(x2(j)+x2(j-1))/Nh)-...
      (p4(j)-h2*k24)*(MIm+(0.5*(u(j)+u(j-1))));
k35 = ((p2(j)-h2*k22)-(p1(j)-h2*k21))*B*Bmh*(0.5*(x1(j)+x1(j-1))/Nh)-...
      (p5(j)-h2*k25)*(MIm+(0.5*(u(j)+u(j-1))));

k41 = ((p2(j)-h*k32)-(p1(j)-h*k31))*B*Bmh*(x5(j-1)/Nh)-(p1(j)-h*k31)*MIh;
k42 = ((p5(j)-h*k35)-(p4(j)-h*k34))*B*Bhm*(x4(j-1)/Nh)-(p2(j)-h*k32)*...
      (ETAh+MIh)+((p3(j)-h*k33)*ETAh-GD);
k43 = -(p3(j)-h*k33)*MIh;
k44 = ((p5(j)-h*k35)-(p4(j)-h*k34))*B*Bhm*(x2(j-1)/Nh)-(p4(j)-h*k34)*...
      (MIm+u(j-1));
k45 = ((p2(j)-h*k32)-(p1(j)-h*k31))*B*Bmh*(x1(j-1)/Nh)-(p5(j)-h*k35)*...
      (MIm+u(j-1));

p1(j-1) = p1(j)-(h/6)*(k11+2*k21+2*k31+k41);
p2(j-1) = p2(j)-(h/6)*(k12+2*k22+2*k32+k42);
p3(j-1) = p3(j)-(h/6)*(k13+2*k23+2*k33+k43);
p4(j-1) = p4(j)-(h/6)*(k14+2*k24+2*k34+k44);
p5(j-1) = p5(j)-(h/6)*(k15+2*k25+2*k35+k45);
end

u1 = min(max(0, (p4.*x4+p5.*x5)/(2*GS)),1);
u = 0.5*(u1 + oldu);

temp1 = delta*sum(abs(u))-sum(abs(oldu-u));

temp21 = delta*sum(abs(x1))-sum(abs(oldx1-x1));
temp22 = delta*sum(abs(x2))-sum(abs(oldx2-x2));
temp23 = delta*sum(abs(x3))-sum(abs(oldx3-x3));
temp24 = delta*sum(abs(x4))-sum(abs(oldx4-x4));
temp25 = delta*sum(abs(x5))-sum(abs(oldx5-x5));

temp31 = delta*sum(abs(p1))-sum(abs(oldp1-p1));
temp32 = delta*sum(abs(p2))-sum(abs(oldp2-p2));
temp33 = delta*sum(abs(p3))-sum(abs(oldp3-p3));
temp34 = delta*sum(abs(p4))-sum(abs(oldp4-p4));
temp35 = delta*sum(abs(p5))-sum(abs(oldp5-p5));

```

```
===== dengue.m =====  
    test = min(temp1, min(temp21, min(temp22, min(temp23, min(temp24, ...  
        min(temp25, min(temp31, min(temp32, min(temp33, min(temp34, ...  
        temp35)))))))));  
end  
  
y(1,:) = t;  
y(2,:) = x2;  
y(3,:) = u;  
  
figure(1)  
plot(y(1,:),y(2:,:)/Nh)  
figure(2)  
plot(y(1,:),y(3,:))
```